

Faculdade de Engenharia da Universidade do Porto



**Visualização de conteúdos multimédia num
ambiente de realidade virtual**

Joel Rocha de Oliveira

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Eletrotécnica e de Computadores
Major de Telecomunicações, Eletrónica e Computadores

Orientador: Prof^o. Eurico Carrapatoso
Coorientadora: Prof^a. Paula Viana

13 de fevereiro de 2018

© Joel Rocha de Oliveira, 2018

Resumo

A presente dissertação aborda a temática do consumo de conteúdos multimédia e de como a evolução tecnológica pode melhorar a experiência dos utilizadores. Atualmente, existe uma enorme facilidade em disponibilizar conteúdos profissionais e pessoais em diversas plataformas *online*; no entanto, o aparecimento de novas tecnologias, juntamente com o aumento das expectativas dos utilizadores, faz com que comecem a surgir novos serviços que possibilitam experiências mais atraentes. Uma das tecnologias que tem gerado maior interesse nos últimos anos é a realidade virtual. Este projeto pretende explorar a utilização desta tecnologia na visualização de conteúdos multimédia, possibilitando uma experiência num ambiente imersivo e interativo.

Nesta dissertação, primeiramente, é efetuado um estudo sobre como funciona a visão e a perceção, bem como sobre as metodologias de visualização existentes. É, ainda, realizado o levantamento das tecnologias relacionadas com a área da visualização e grafismo *web*.

De seguida, faz-se uma observação mais pormenorizada do problema, e explica-se a sua origem e uma forma como pode ser solucionado.

Posteriormente, apresenta-se a ferramenta que se utilizou no desenvolvimento do protótipo da solução proposta, explicando as suas funções básicas e as suas características.

Por último, procede-se a uma apresentação da aplicação desenvolvida, onde se fundamentam as escolhas feitas sobre o *design*, pormenoriza-se cada elemento e explicita-se o seu funcionamento e as principais características. Apresenta-se seguidamente o questionário realizado com o objetivo de validar o trabalho desenvolvido.

Abstract

The present dissertation is on the theme of the consumption of multimedia contentes and on how the technological evolution can improve the users' experience.

Nowadays, it is very easy to provide professional and personal contents in several online platforms. However, the appearing of new technologies, together with the raise of the users' expectations, is leading to the forthcoming of new services that make it possible to have more attractive experiences. One of the technologies that has created more interest lately is virtual reality. This project intends to explore the use of this technology in the viewing of multimedia contents, enabling the experience in an immersive and interactive ambience.

In this document, we firstly study on how the vision and perception work, as well as on the existing viewing methodologies. We also make a survey on the technologies that are related to the area of visualization and web graphics.

Then, we observe the problem in detail; we explain its origin and we also suggest a way to solve it.

After that, we present the tool that we used in the development of the prototype of the suggested solution and we explain its basic functions and characteristics.

Finally, we present the application that has been developed and we justify the choices on the design, we specify every element and we explain its operation and its main characteristics. At this moment, we present the result of the survey in order to validate our work.

Agradecimentos

Ao Professor Eurico Carrapatoso, pela confiança que depositou em mim, pela orientação e paciência ao longo do ano.

À minha família, por todo o apoio e por possibilitarem este meu percurso de aprendizagem.

Aos meus amigos e colegas que me acompanharam nesta jornada.

À minha namorada, por toda a ajuda, pelo apoio e incentivo em todos os momentos.

O meu Muito Obrigado!

*“Even when the information is terrible,
the visual can be quite beautiful.”*

David McCandless

Índice

Resumo	iii
Abstract.....	v
Agradecimentos	vii
Índice.....	xi
Lista de figuras	xiii
Lista de tabelas	xv
Abreviaturas e símbolos	xvii
1. Introdução	1
1.1 Contexto.....	1
1.2 Motivação	2
1.3 Objetivos	3
2. Estado da arte.....	5
2.1 Visualização e percepção	5
2.2 3D <i>versus</i> 2D	7
2.3 Realidade virtual na <i>web</i>	8
2.4 Formas visuais de apresentação de informação	9
2.5 Projetar a visualização de informação	10
2.6 APIs de gráficos 3D	10
2.7 APIs de gráficos 2D	11
2.8 Bibliotecas de desenvolvimento com WebGL	12
2.9 Bibliotecas para visualização de grafos	15
2.10 Trabalho relacionado.....	16
3. Análise do problema	19
3.1 Definição do problema	19
3.2 Solução proposta	20
3.3 Síntese	22
4. A-Frame	23
4.1 O que é o A-Frame?.....	23
4.2 O básico	24
4.3 Características	28

4.4	Comunidade	30
4.5	Síntese	30
5.	A aplicação: MondoVR	31
5.1	Introdução ao MondoVR.....	31
5.2	Arquitetura	32
5.3	<i>Web design</i>	33
5.4	Conteúdos.....	37
5.5	Componentes	38
5.6	Entidades e primitivas	42
5.7	Funcionamento.....	49
5.8	Problemas conhecidos	51
5.9	Discussão de resultados.....	52
6.	Conclusões e trabalho futuro	57
6.1	Conclusões	57
6.2	Trabalho futuro	58
	Referências	59

Lista de figuras

Figura 1 - O processo de visualização	6
Figura 2 - Modelo de três fases do processo de visualização de informação	6
Figura 3 - A evolução da metodologia de visualização	8
Figura 4 - Tipos de gráficos: Histograma, Barras, Círculo e Linhas	9
Figura 5 - Visualização de um desenho de grafos ou diagrama em rede (baseado em força) ...	9
Figura 6 - Imagem estática da construção de uma cena básica em A-Frame.....	13
Figura 7 - Exemplo de gráfico para o artista “Radiohead”	17
Figura 8 - Exemplo de estrutura (3pqr.pdb) disponibilizada na aplicação <i>web</i>	17
Figura 9 - Página inicial (à esquerda) e página da comunidade Kew (à direita)	18
Figura 10 - <i>Mockup</i> concebido no início da dissertação.	20
Figura 11 - Os principais <i>headsets</i> de realidade virtual disponíveis no mercado	21
Figura 12 - VR Shinecon 3D VR Glasses Headset	21
Figura 13 - A-Frame, CDN build (versão 0.7.0).	24
Figura 14 - Código do “Hello, World!” do A-Frame.	24
Figura 15 - Cenário do “Hello, World!” do A-Frame.	25
Figura 16 - Utilização do Glitch	26
Figura 17 - Utilização do A-Frame Inspector.	27
Figura 18 - A-Painter (menu principal do controlador).....	27
Figura 19 - Diagrama da arquitetura da aplicação.	32
Figura 20 - Diagrama sobre o funcionamento da aplicação na visualização de um vídeo.	33
Figura 21 - Plano afastado da aplicação MondoVR.	34

Figura 22 - Organização em 2D dos principais elementos da aplicação.	35
Figura 23 - Os diferentes estados do cursor: à esquerda, o cursor encontra-se sobre um objeto não clicável; ao centro, o cursor encontra-se sobre um objeto clicável; e à direita, o cursor encontra-se no estado iminente que antecede o clique.	37
Figura 24 - Exemplo da utilização do componente <i>template</i>	38
Figura 25 - Excerto do código da aplicação que representa a utilização da componente <i>template</i>	39
Figura 26 - Exemplo da utilização do componente <i>layout</i>	39
Figura 27 - Excerto do código da aplicação que representa a utilização da componente <i>layout</i>	40
Figura 28 - Exemplo da utilização do componente <i>event-set</i>	40
Figura 29 - Atributos da primitiva <i><a-sky></i>	42
Figura 30 - Atributos da primitiva <i><a-curvedimage></i>	43
Figura 31 - Resultado visual de uma entidade <i>plane</i> e legenda respetiva.	44
Figura 32 - Código das entidades relacionadas com o ecrã.	45
Figura 33 - Resultado visual das entidades relacionadas com o ecrã.	45
Figura 34 - Código das entidades e primitivas relacionadas com as informações sobre os vídeos.	46
Figura 35 - Resultado visual das entidades e primitivas relacionadas com as informações sobre os vídeos.	46
Figura 36 - Código da entidade relacionada com a categoria “Desporto”.	47
Figura 37 - Resultado visual das entidades relacionadas com as categorias.	47
Figura 38 - Código das entidades relacionadas com a câmara e com o cursor.	48
Figura 39 - Botão para entrar no modo <i>fullscreen</i> ou no modo realidade virtual com um <i>headset</i>	49
Figura 40 - Funcionamento do sistema de reprodução de vídeos.	49
Figura 41 - Imagem estereoscópica criada no modo realidade virtual.	50
Figura 42 - Gráfico das respostas à pergunta 1.	53
Figura 43 - Gráfico das respostas à pergunta 2.	54
Figura 44 - Gráfico das respostas à pergunta 3.	54
Figura 45 - Gráfico das respostas à pergunta 4.	55
Figura 46 - Gráfico das respostas à pergunta 6.	56
Figura 47 - As 3 fases de trabalho da aplicação idealizada.	58

Lista de tabelas

Tabela 1 - Vantagens e desvantagens 2D.....	7
Tabela 2 - Vantagens e desvantagens 3D.....	7
Tabela 3 - Comparação das bibliotecas de desenvolvimento com WebGL (parte 1/2)	14
Tabela 4 - Comparação das bibliotecas de desenvolvimento com WebGL (parte 2/2)	14
Tabela 5 - Comparação das bibliotecas para visualização de grafos	16

Abreviaturas e símbolos

Lista de abreviaturas (ordenadas por ordem alfabética)

2D	Duas Dimensões
3D	Três Dimensões
AGPL	<i>Affero General Public License</i>
API	<i>Application Programming Interface</i>
BSD	<i>Berkeley Software Distribution</i>
CDN	<i>Content Delivery Network</i>
COLLADA	<i>COLLABorative Design Activity</i>
CSS	<i>Cascading Style Sheets</i>
D3	<i>Data-Driven Documents</i>
ES	<i>Embedded Systems</i>
FBX	<i>Filmbox</i>
glTF	<i>GL Transmission Format</i>
GPU	<i>Graphics Processing Unit</i>
HTC	<i>High Tech Computer Corporation</i>
HTML	<i>HyperText Markup Language</i>
JS	<i>JavaScript</i>
MIT	<i>Massachusetts Institute of Technology</i>
NPM	<i>Noxious Plant Miasma</i>
OBJ	<i>Object</i>
OpenGL	<i>Open Graphics Library</i>
PDB	<i>Protein Data Bank</i>
RAMA	<i>Relational Artist MApps</i>
STL	<i>Stereolithography</i>
SVG	<i>Scalable Vector Graphics</i>
VR	<i>Virtual Reality</i>
WebVR	<i>Web Virtual Reality</i>

Capítulo 1

Introdução

1.1 Contexto

Atualmente, aceder a conteúdos multimédia torna-se cada vez mais fácil devido à enorme disponibilidade de conteúdos profissionais e pessoais em plataformas *online*. O avanço da tecnologia, as expectativas dos utilizadores e os resultados das atividades de investigação fizeram com que a quantidade de conteúdos fornecidos aumentasse significativamente [1].

Deste modo, é expectável que comecem a surgir conteúdos de diferentes formatos numa perspetiva de levar o utilizador a um novo nível de absorção de informação e, ainda, a envolver-se com os serviços fornecidos.

O conteúdo audiovisual possui como potencialidade a capacidade de causar um impacto significativo nas emoções dos utilizadores e no seu sentimento de presença. Esse conteúdo multimodal e de ambiente imersivo é conhecido por provocar emoções muito mais marcantes em comparação com o tradicional conteúdo textual. A imersão pode ser influenciada por variáveis sensoriais, efeitos envolventes, mas também pela participação e interação social na cadeia dos *media*, aumentando assim o sentimento de pertença e o envolvimento com o serviço, aplicação, ou informação que está a ser fornecida [1].

Um elemento importante na visualização de informação é a percepção. Esta consiste no processo ou na capacidade de conhecer e compreender o que nos rodeia através da interpretação, seleção e organização de diferentes tipos de informações. Todas as percepções implicam estímulos do sistema nervoso central, que são o resultado da estimulação dos nossos órgãos sensoriais [2]. O estudo da percepção por parte de psicólogos e neurocientistas tem evoluído bastante ao longo das últimas três décadas o que possibilita uma forte reflexão acerca do seu impacto na visualização de informação [3].

A organização desta informação assume, igualmente, um papel fundamental na forma como é visualizada. Ao longo dos tempos foi possível observar uma evolução no modelo visual mais usado para representar diferentes temáticas. Desde as estruturas hierárquicas, onde tudo o

que nos rodeia tinha de ter uma ordem natural, passando pelos esquemas ramificados de uma árvore, que ainda hoje são usados, por exemplo, para representar as gerações da família. No entanto, nos últimos anos tem surgido mais frequentemente um novo modelo visual. As ligações em rede têm provado constituir uma representação mais rigorosa e com uma enorme possibilidade de adaptar diversas formas e tipologias. Este modelo visual em forma de teia promete tornar-se na sintaxe de uma nova linguagem [4].

1.2 Motivação

Estes são tempos empolgantes para o mundo do *design* da visualização. A tecnologia computacional utilizada para produzir matéria visual atingiu um estágio em que informação visual 3D, interativa e sofisticada, pode ser criada a partir de qualquer computador portátil ou *tablet*. A tendência para se usar informação visual está a crescer cada vez mais, e há uma explosão de novas técnicas de visualização a serem inventadas para auxiliar na necessidade de analisar quantidades enormes e complexas de dados e informação. Esta fase criativa, por norma não é duradoura. Habitualmente, com o surgimento de uma nova tecnologia, apenas existe um curto período de manifestação criativa até que as forças padronizadas transformem o que é novo em convencional. É indiscutível que muitas das técnicas de visualização que estão a emergir atualmente, tornem-se em ferramentas de rotina num futuro próximo. Até mesmo projetos mal concebidos podem tornar-se em padrões da indústria. Projetar para a perceção pode ajudar a evitar tais erros. Se aproveitarmos o conhecimento acumulado sobre o modo como funciona a perceção, talvez seja possível tornar a visualização muito mais transparente no mundo da informação [3].

Mas porquê o interesse pela visualização? Porque o sistema visual humano é um detetor de padrões de enorme poder e subtilidade. O olho e o córtex visual do cérebro formam um gigantesco processador paralelo que fornece o canal de maior largura de banda para o sistema cognitivo humano. Na realidade, nós adquirimos mais informação através da visão do que através de todos os outros sentidos combinados. No entanto, o sistema visual tem as suas próprias regras. Podemos ver facilmente padrões apresentados de certas maneiras, contudo se estes são apresentados de maneira diferente, tornam-se invisíveis. Se pudermos entender como funciona a perceção, o nosso conhecimento pode ser traduzido em normas para a forma como exibimos a informação. Seguindo regras baseadas na perceção, podemos apresentar os nossos dados de tal forma que os padrões importantes e informativos se destaquem, caso contrário os nossos dados serão incompreensíveis [3].

Quando estamos a visualizar informação, nós transformamo-la numa paisagem em que podemos explorar com os nossos olhos, uma espécie de mapa de informações. E quando estamos perdidos em informações, um mapa de informações é algo bastante útil [5].

Nos dias de hoje, quando pesquisamos ou navegamos *online*, somos inundados com milhares de conteúdos que gostaríamos que fossem exibidos de forma sofisticada e navegável.

1.3 Objetivos

É amplamente reconhecido que os conteúdos multimédia interativos desempenham um papel cada vez mais importante não só na indústria cultural, mas em quase todos os aspetos da vida socioeconómica e lúdica. Ambientes virtuais e imersivos, com a capacidade de envolver múltiplos sentidos, estão igualmente a surgir [1]. Pretende-se explorar uma nova forma de navegação de conteúdo multimédia que utilize como base novos paradigmas de análise visual e que proporcione um conceito mais atraente e imersivo de consumo de conteúdo.

O objetivo **principal** para a dissertação é conceber e desenvolver uma aplicação visual *web* interativa e sofisticada.

Os objetivos **secundários** são:

1. Estudo das tecnologias atualmente existentes na área da visualização e grafismo *web*.
2. Definição de uma API multiplataforma de baixo nível e de bibliotecas de alto nível para animações *web*.
3. Conceção e desenvolvimento de uma aplicação *web* teste que utilize as tecnologias escolhidas.
4. Recolha de um conjunto de conteúdos multimédia.
5. Integração dos conteúdos multimédia na aplicação *web* e respetivos testes e validações.

1.4 Estrutura do documento

Este documento encontra-se dividido em seis capítulos.

O capítulo 1 apresenta o contexto, a motivação e os objetivos, principais e secundários, da presente dissertação.

No capítulo 2 é efetuado o levantamento bibliográfico, estado da arte, relativo às ferramentas e tecnologias atualmente existentes na área da visualização e grafismo *web*.

No capítulo 3 analisa-se o problema que se pretende solucionar nesta dissertação e identifica-se as tecnologias e ferramentas utilizadas.

No capítulo 4 é dissecada a ferramenta escolhida, com o objetivo de permitir um melhor entendimento sobre o seu funcionamento, importante para a compreensão de algumas sessões do capítulo seguinte.

O capítulo 5 apresenta detalhadamente a aplicação desenvolvida para a visualização de conteúdos multimédia.

Por último, no capítulo 6 estão presentes as conclusões resultantes desta dissertação e os possíveis trabalhos a desenvolver futuramente.

Capítulo 2

Estado da arte

Existe uma quantidade enorme de elementos que afetam a visualização de algo. O ambiente, a luminosidade, o brilho, o contraste, a cor, a saliência visual, os padrões estáticos e de movimento, a percepção do espaço, entre outros. Dado o pouco que se assemelha ao mundo em que vivemos, é surpreendente como uma superfície retangular plana consegue ser tão bem-sucedida na apresentação de informação, como é o caso de um monitor de um computador [3].

Neste capítulo são abordados alguns temas como o processo da visualização e percepção, a comparação do 2D com o 3D e algumas formas visuais de apresentação de informação. Debate-se ainda as tecnologias atualmente existentes para o desenvolvimento de aplicações *web* animadas.

2.1 Visualização e percepção

O processo de visualização de informação envolve quatro fases, combinadas em vários ciclos de realimentação, como se pode ver na Figura 1. As quatro fases consistem:

- Na recolha e armazenamento de informação;
- Numa etapa de pré-processamento para transformar os dados em algo mais fácil de manipular. Geralmente existe uma redução de dados para revelar os aspetos seleccionados;
- No mapeamento dos dados seleccionados para uma representação visual, que é conseguida através de algoritmos computacionais que geram a imagem no ecrã;
- No processamento visual e cognitivo de informação pelo percetor.

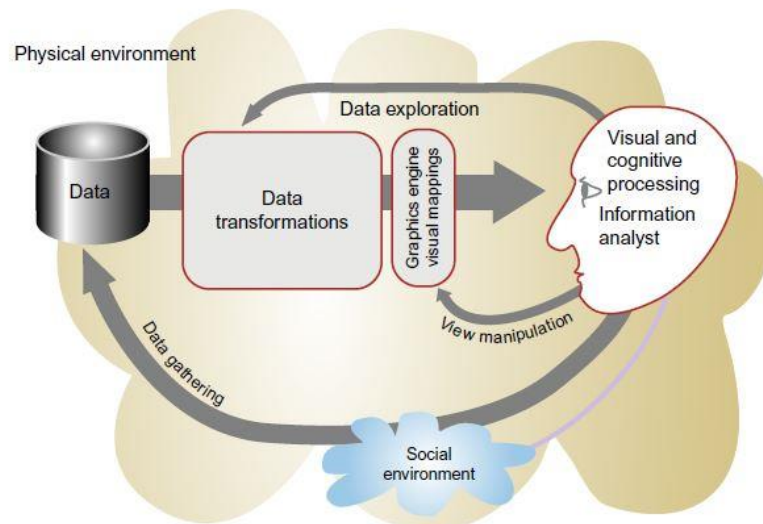


Figura 1 - O processo de visualização [3].

No entanto, existem muitos subsistemas envolvidos na visão e, por isso, deve-se ter cuidado para não se generalizar em demasia na análise de dados. Apesar disso, um modelo mais simples, como o da Figura 2, é útil pois apresenta uma visão esquemática mais ampla de um modelo de percepção em 3 fases [3].

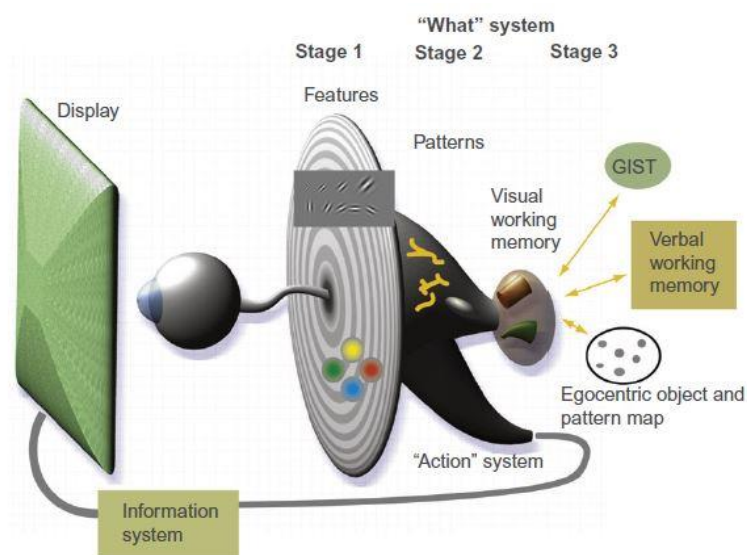


Figura 2 - Modelo de três fases do processo de visualização de informação [3].

Na fase 1, as informações são processadas em paralelo para extrair características básicas do ambiente. Na fase 2, os processos ativos de percepção de padrões extraem estruturas e segmentam a cena visual em diferentes regiões de cor, textura e de padrões de movimento. Na fase 3, os mecanismos ativos de atenção reduzem a informação para apenas alguns objetos mantidos na memória de trabalho visual com o propósito de formar a base do pensamento visual [3].

2.2 3D versus 2D

O interesse pelo 3D e pela realidade virtual tem aumentado significativamente nos últimos anos e com isso o aparecimento de diversos sistemas imersivos pelas principais marcas da tecnologia, como por exemplo Google, Samsung, Facebook, Microsoft, HTC e Sony.

Criar um ambiente de visualização 3D é consideravelmente mais difícil do que criar um sistema 2D com capacidades semelhantes. Faltam ainda regras de *design* para ambientes 3D e muitas técnicas de interação competem para serem a principal escolha. O principal argumento para a ascensão de sistemas de visualização 3D e interfaces de utilizador 3D em geral, refere-se à nossa existência ser num mundo 3D e que os nossos cérebros evoluíram para reconhecer e interagir dentro de 3D [3].

No entanto, passar de 2D para 3D acrescenta muito menos informação visual do que se poderia supor. A maior parte dos nossos sistemas visuais de percepção trabalham em 2D e não em 3D. Estima-se que seremos capazes de fundir estereoscopicamente imagens que difiram até 10%. Por essa razão, mesmo utilizando uma visualização 3D, é fundamental que se perceba a informação exibida num plano de imagem 2D [3].

Tabela 1 - Vantagens e desvantagens 2D [6].

Vantagens 2D	Desvantagens 2D
Menor custo de produção	Menor preferência
Rapidez de produção	Menor interesse
Simplicidade	Baixa reutilização

Tabela 2 - Vantagens e desvantagens 3D [6].

Vantagens 3D	Desvantagens 3D
Maior realismo	Complexidade
Maior reutilização	Imaginação limitada à tecnologia
Maior qualidade	Baixo rácio esforço/recompensa

2.3 Realidade virtual na *web*

Ao longo dos tempos, a metodologia de visualização foi sempre evoluindo, especialmente nas últimas décadas devido ao desenvolvimento tecnológico (ver na Figura 3). A Realidade virtual é uma tecnologia que adiciona as componentes de imersão e interatividade e está a crescer rapidamente nos últimos anos. Já é possível ter experiências virtuais pela *web* através de páginas que permitam WebVR e com a utilização de um *headset* de realidade virtual. WebVR é uma API de JavaScript experimental que fornece suporte para dispositivos de realidade virtual num navegador. O Mozilla foi pioneiro neste campo ao desenvolver várias ferramentas que permitiram levar a realidade virtual para a *web* [7].

O WebVR tem o potencial de permitir experiências incríveis. Imagine-se o Wikipedia onde, em vez de se ler uma página sobre história, seria possível “viver” essa história com todo o impacto audiovisual inerente. Imagine-se o Amazon, que se poderia tornar num shopping virtual onde se poderia experimentar roupas por exemplo [8].

O potencial existe, mas ainda falta um longo caminho para se chegar a uma *web* interligada onde se possa navegar em realidade virtual. Neste momento apenas existem páginas isoladas onde os programadores procuram levar o conteúdo 2D para um *headset* de realidade virtual. Um grande problema para o sucesso da realidade virtual na *web* são as empresas, tais como Google e Apple, que têm o seu próprio ecossistema e querem aplicações nativas pois assim conseguem controlar toda a cadeia de valor [8] [9].

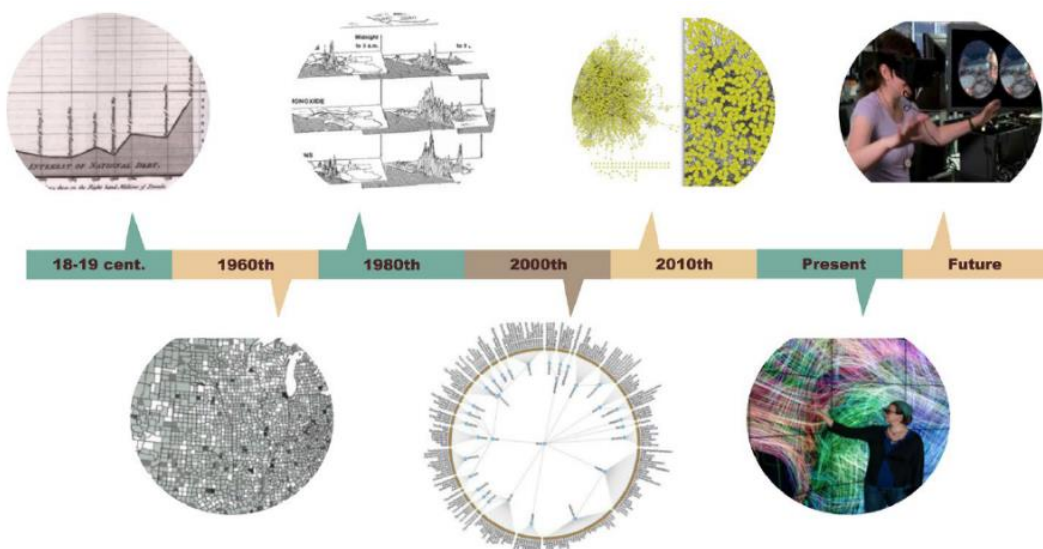


Figura 3 - A evolução da metodologia de visualização [10].

2.4 Formas visuais de apresentação de informação

A visão desempenha um papel fundamental na forma como adquirimos informação. Talvez seja por isso que uma das formas mais comuns de apresentação de informação seja através de gráficos. Estes permitem uma maior facilidade na interpretação e compreensão de grandes quantidades de dados e fazem com que padrões e tendências sejam muito mais facilmente visíveis. Existem vários tipos de gráficos, sendo os mais utilizados os histogramas, os gráficos de barras, os gráficos de círculo e os gráficos de linhas (ver Figura 4) [11].

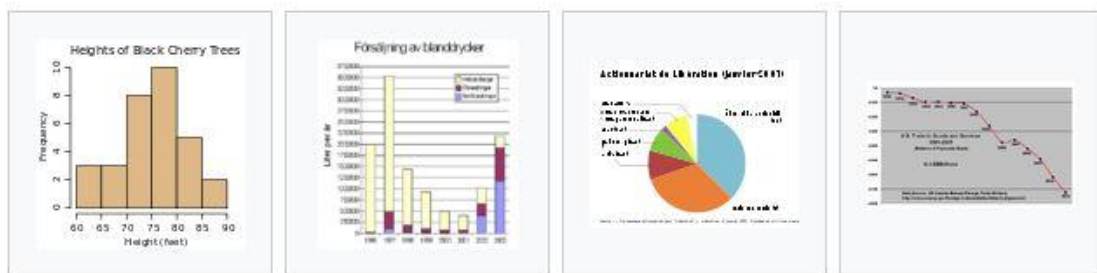


Figura 4 - Tipos de gráficos: Histograma, Barras, Círculo e Linhas [11].

Contudo, uns são mais adequados para um certo tipo de informação do que outros. No que diz respeito à visualização de informação, os mapas apresentam um melhor relacionamento visual e conseguem fornecer informação extra para um propósito específico. Exemplos disso são os mapas de calor, os mapas de árvore, os mapas conceituais e o desenho de grafos (ou diagrama em rede - ver Figura 5) [11] [12].

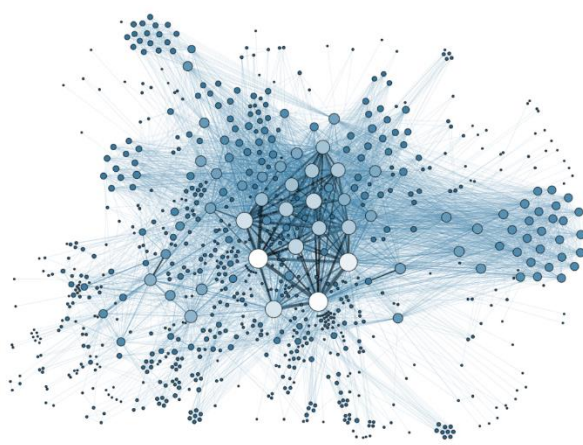


Figura 5 - Visualização de um desenho de grafos ou diagrama em rede (baseado em força) [13].

2.5 Projetar a visualização de informação

Existem várias maneiras de projetar a visualização de informação e por isso seria impossível providenciar instruções passo a passo para tantas possibilidades. No entanto, o processo pelo qual se chega ao melhor produto final possível é frequentemente o mesmo. Começa-se por definir o problema, a informação a ser representada, as dimensões necessárias para representar a informação, as suas relações e o tipo de interação necessária [14] [15].

Um dos passos mais importantes é a análise de como a informação se relaciona entre si. As estruturas mais comuns são:

- 1) **Relações lineares.** Onde a informação pode ser exibida em formato linear, como tabelas, vetores, etc.
- 2) **Relações temporais.** Onde a informação muda ao passar do tempo.
- 3) **Relações espaciais.** Onde a informação se relaciona com o mundo real (mapas, etc.).
- 4) **Relações hierárquicas.** Informação que se relaciona com posições de uma hierarquia definida.
- 5) **Relações em rede.** Onde a informação se relaciona com outras entidades dentro da mesma informação.

2.6 APIs de gráficos 3D

2.6.1 WebGL

O WebGL é uma API em JavaScript, multiplataforma e de baixo nível para renderização de gráficos 3D e disponível a partir do elemento Canvas do HTML5. Pode ser implementado numa aplicação *web* sem a necessidade de *plug-ins* no navegador. O WebGL é baseado no OpenGL ES: o WebGL 1.0 expõe o conjunto de recursos do OpenGL ES 2.0 e o WebGL 2.0 exhibe a API do OpenGL ES 3.0 [16] [17] [18].

O WebGL está integrado completamente em todas as normas dos navegadores *web*, permitindo a aceleração do GPU para o processamento de imagens e efeitos. É ainda suportado por uma larga maioria dos navegadores *web* modernos com os principais fornecedores, Apple (Safari), Google (Chrome), Microsoft (Edge) e Mozilla (Firefox), a fazerem parte do *WebGL Working Group* [16] [17].

Esta API pode ser bastante maçadora para se usar diretamente sem o auxílio de algumas bibliotecas utilitárias. Existem diversas bibliotecas de JavaScript que foram desenvolvidas com o propósito de facilitar e fornecer funcionalidades adicionais [17]. Algumas delas serão discutidas na secção 2.8.

O WebGL foi desenvolvido e mantido pelo grupo sem fins lucrativos Khronos Group [16].

2.6.2 Unity 3D

Unity, também conhecido por Unity 3D, não é propriamente uma API, mas sim um motor que tem como alvo diferentes APIs (Direct3D no Windows e Xbox 360; OpenGL no Mac, Windows e Linux; OpenGL ES no Android e iOS; e APIs proprietárias nas consolas de videojogos). Consiste num motor mais direcionado para jogos de navegador *web* devido às suas características de mapeamento de colisão, mapeamento de reflexão, entre muitas outras. Algumas das suas vantagens são a estabilidade e a leveza enquanto que a principal desvantagem é a necessidade de instalação do *Unity Web Player*, um *plug-in* para o navegador, que o utilizador é obrigado a ter [19] [20].

2.6.3 Stage 3D

Stage 3D é uma API do Adobe Flash Player para a renderização de gráficos 3D interativos com aceleração do GPU, para jogos e aplicações em Flash, escritos em ActionScript 3. Não parece ter grandes vantagens em relação a outras APIs com o mesmo propósito e *design*, como o WebGL. O facto de usar uma API não normalizada, de ter uma documentação pobre, de não correr em dispositivos móveis e de não integrar bem com nenhuma das tecnologias *web*, deixa o Stage 3D longe da concorrência [18] [21].

2.6.4 Outras

A tecnologia GPU tem vindo a melhorar e têm surgido novos softwares de APIs para refletir melhor os projetos das GPUs modernas. Estas novas APIs são de um baixo nível de abstração e, devido à sua reduzida sobrecarga, geralmente oferecem melhores desempenhos que o OpenGL. As principais plataformas de tecnologia neste campo são o Direct3D 12 da Microsoft, o Metal da Apple e o Vulkan do Khronos Group. Embora estas tecnologias tenham conceitos de *design* semelhantes, infelizmente nenhuma está disponível em todas as plataformas [22].

2.7 APIs de gráficos 2D

Relativamente às APIs para renderização de gráficos 2D e que sejam baseadas na *web*, existe uma que se destaca: o elemento Canvas do HTML5. Algumas APIs de gráficos 3D, como é o caso do WebGL, suportam também a renderização de gráficos 2D pois utilizam o HTML5 Canvas [17].

2.7.1 HTML5 Canvas

O elemento Canvas é um modelo de baixo nível que faz parte do HTML5 e permite uma renderização dinâmica de formas 2D e de imagens bitmap. Consiste essencialmente numa área editável definida no código HTML com atributos de altura e largura. O código JavaScript pode aceder a esta área através de um conjunto de funções de desenho, permitindo gerar gráficos dinâmicos. Algumas das principais utilizações do Canvas incluem a criação de gráficos animados, de jogos, e de composição de imagem [23].

2.8 Bibliotecas de desenvolvimento com WebGL

Existem dezenas de bibliotecas que usam WebGL. Apresentam-se apenas algumas das mais populares.

2.8.1 Three.js

Three.js é uma biblioteca JavaScript de alto nível usada para criar e mostrar gráficos 3D animados num navegador *web*. É uma das bibliotecas mais populares dada a forma fantástica como abstrai toda a confusão do WebGL e permite a quem desenvolve, concentrar-se na criação de cenas 3D [24] [25].

Para além de uma boa documentação, o Three.js possui uma larga comunidade e suporte na *web*. O código-fonte está alojado num repositório no GitHub [26].

2.8.2 Babylon.js

Atualmente, o Babylon.js é a biblioteca JavaScript mais popular na comunidade *web*, a par da Three.js. Enquanto que o Three.js foi desenvolvido como uma ferramenta para animações *web* de uso geral, o Babylon.js foi projetado para ter uma abordagem mais direcionada para o desenvolvimento de jogos *web*, utilizando um extenso motor de física para a deteção de colisões e outras complexidades envolvidas em gráficos 3D [27].

O Babylon.js conta também com um vasto suporte na *web* e o seu código-fonte também está alojado num repositório no GitHub [28].

2.8.3 A-Frame

A-Frame, uma das mais recentes bibliotecas de código aberto, permite a rápida criação de cenas 3D e de realidade virtual na *web*, tendo sido desenvolvida pela equipe de VR do Mozilla. O A-Frame simplifica o desenvolvimento de WebGL de tal forma que praticamente qualquer desenvolvedor *web* ou *designer* pode começar a criar cenas em minutos (ver exemplo na Figura 6). Isto é possível uma vez que integra a biblioteca Three.js e empacota-a em pequenos elementos puros de HTML. Esses elementos permitem criar uma cena 3D da mesma maneira que se cria qualquer outra página da *web*. Para além disso, pode-se usar APIs JavaScript para manipular cenas A-Frame e adicionar lógica, comportamento e funcionalidade [29] [25].

Bibliotecas como React, Vue.js, d3.js, jQuery, ou Angular, funcionam bem sobre A-Frame, uma vez que foram construídas em cima da noção da manipulação de HTML simples [30].

Na página *web* oficial do A-Frame é possível encontrar diversos guias de apoio bem como ferramentas e componentes como o A-Frame Inspector, A-Painter e o A-Frame Registry [30].

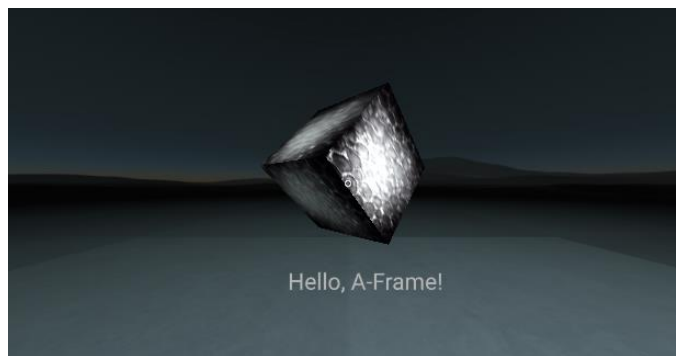


Figura 6 - Imagem estática da construção de uma cena básica em A-Frame [30].

2.8.4 PlayCanvas

À semelhança do Babylon.js, o PlayCanvas é uma biblioteca JavaScript orientada para o desenvolvimento de jogos *web* e suportada pelos principais navegadores. A principal diferença está nas ferramentas que oferece. O PlayCanvas possui uma poderosa plataforma proprietária de criação, alojada na nuvem, que permite a edição simultânea de vários computadores através do navegador *web*. A sua vantagem é também a sua desvantagem. Por ser tão focado no seu editor, pode não fazer muito sentido usar a biblioteca de forma isolada, fazendo do PlayCanvas uma opção mais adequada para a criação de experiências únicas. A comunidade é boa, mas é principalmente focada no seu editor [31] [32].

O PlayCanvas é gratuito para projetos públicos, no entanto, para projetos privados, é necessário o pagamento de taxas mensais [33].

2.8.5 Comparação

Tabela 3 - Comparação das bibliotecas de desenvolvimento com WebGL (parte 1/2) [34].

Nome	Scripting	Modelagem	Animação	Áudio integrado	Trabalho em rede	Física integrada
Three.js	JavaScript	Não	Sim	Sim	Sim	Não
Babylon.js	JavaScript TypeScript	Não	Sim	Sim	Não	Sim
A-Frame	JavaScript	Não	Sim	Sim	Não	Não
PlayCanvas	JavaScript	Não	Sim	Sim	Sim	Sim

Tabela 4 - Comparação das bibliotecas de desenvolvimento com WebGL (parte 2/2) [34].

Nome	Independência da nuvem	Versão do WebGL	WebVR	Importa	Exporta	Licença
Three.js	Não	1.0	Sim	FBX, OBJ, STL	OBJ	MIT
Babylon.js	Sim	1.0	Não	OBJ, FBX, STL, Babylon	Todos formatos suportados pelo Blender e 3dsMax	Apache 2.0
A-Frame	Não	1.0	Sim	OBJ, COLLADA, glTF, FBX, three.js	HTML, three.js	MIT
PlayCanvas	Não	1.0	Não	FBX, OBJ	Não	MIT (motor), editor proprietário

2.9 Bibliotecas para visualização de grafos

2.9.1 D3.js

O D3.js é a mais popular e poderosa biblioteca JavaScript, de código aberto, para a criação de vários tipos de gráficos e para a visualização de dados em geral. É flexível e funciona bem com as tecnologias *web* (HTML, CSS, SVG). Por ser multifacetado, o D3 requer mais trabalho para o mesmo resultado comparado com outras bibliotecas. Possui uma grande comunidade, o que se traduz em mais recursos disponíveis [35] [36] [37].

2.9.2 Sigma.js

O Sigma.js é uma biblioteca JavaScript mais especializada em desenhar grafos. Nessa área, é mais rápido, mais interativo e lida melhor com grafos de maior tamanho do que o D3. O Sigma fornece vários recursos, como por exemplo o suporte para WebGL e Canvas [38].

2.9.3 Vis.js

O Vis.js é uma biblioteca dinâmica e baseada no navegador *web*. Foi projetada para ser fácil de utilizar, para lidar com grandes quantidades de dados e para permitir a manipulação e interação com os dados. Suporta os principais tipos de gráficos, incluindo o desenho de grafos, e suporta também 3D, embora apenas nos gráficos de barras [37] [39].

2.9.4 Alchemy.js

O Alchemy.js é uma biblioteca de desenho de grafos construída quase inteiramente em D3. Simples e minimalista, o Alchemy requer pouco código para personalizar e gerar grafos visto que a maioria das implementações são substituições da configuração padrão, em vez da implementação direta via JavaScript [40].

2.9.5 Comparação

Tabela 5 - Comparação das bibliotecas para visualização de grafos [37].

Nome	Tipos de gráficos	Tipos de gráficos de barra	Visualização em grafos (rede)	3D	Complexidade	Licença
D3.js	Todos	Todos	Sim	Não	Alta	BSD-3
Sigma.js	Rede	Nenhum	Sim	Não	Baixa	MIT
Vis.js	Linha Cronograma Dispersão Área Gantt Rede Agrupado	Empilhado Negativo	Sim	Sim (alguns)	Baixa	MIT Apache 2.0
Alchemy.js	Rede	Nenhum	Sim	Não	Baixa	AGPLv3

2.10 Trabalho relacionado

Visualizações de dados e de informação multimédia são usadas em diversas áreas desde jornalismo, arte, redes sociais, análise de dados e estatística. De seguida apresentam-se alguns exemplos de aplicações encontrados em artigos científicos.

2.10.1 - RAMA

O RAMA (*Relational Artist Maps*) é uma interface 2D simples e eficiente para navegar através de redes de artistas de música, desenvolvido em Processing. O RAMA é construído sobre um conjunto de dados de similaridade de artistas e *tags* definidas pelos utilizadores em relação a 583.000 artistas reunidos do Last.fm. Estes dados de terceiros, publicamente disponíveis, sobre similaridade de artistas e *tags* de artistas são usados para produzir uma visualização de relações de artistas. O RAMA fornece duas camadas simultâneas de informações: (i) um gráfico construído a partir de dados de similaridade de artistas e (ii) rótulos sobrepostos contendo *tags* definidas pelos utilizadores. Contrariamente das ferramentas existentes de visualização de redes de artistas, o protótipo proposto enfatiza pontos em comum, bem como as principais

2.10.3 - Silk River Web App

Como parte de um programa anual que marca o 70º aniversário da independência da Índia e das relações culturais com o Reino Unido, Kinetika criou o Silk River, um projeto que explora a relação única entre Londres e Kolkata através do intercâmbio entre comunidades ao longo do estuário do Tâmesa e o rio Hooghly da Índia [44] [45].

O Silk River Web App é uma aplicação simples 2D que reúne conteúdo multimédia artístico destas comunidades e exibe-o ao longo de uma imagem alusiva. Dentro da página de cada comunidade encontramos uma apresentação informativa, galeria de imagens, vídeos e ficheiros áudio (ver Figura 9).

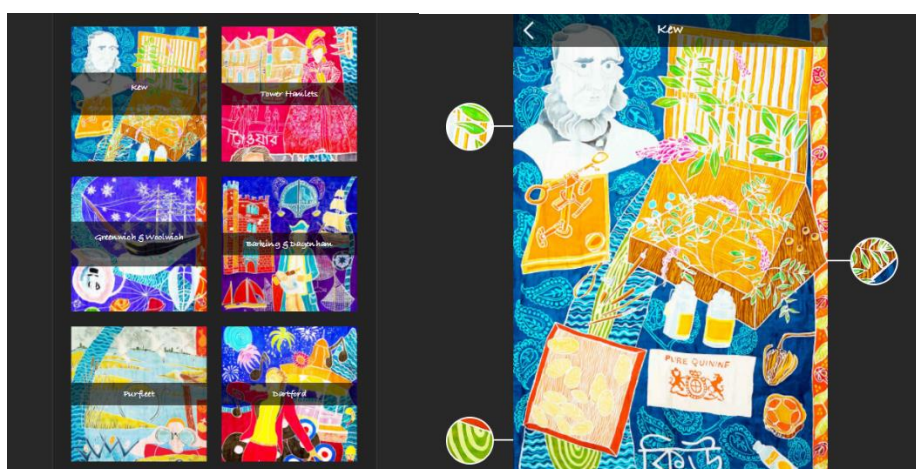


Figura 9 - Página inicial (à esquerda) e página da comunidade Kew (à direita) [46].

Capítulo 3

Análise do problema

Neste capítulo apresenta-se o problema que deu origem a esta dissertação tendo como objetivo dar ao leitor uma visão geral sobre o tema. Pretende-se também conduzir o leitor pelo processo de análise que foi feito até chegar à solução proposta bem como a fundamentação da escolha das ferramentas e tecnologias.

3.1 Definição do problema

Podemos dizer que, nos dias de hoje, a Internet está a tornar-se numa “Internet das pessoas” muito devido à explosão das redes sociais e ao seu impacto no quotidiano das pessoas. Capturar, produzir, partilhar e aceder a conteúdos e informações das próprias perspetivas e experiências dos utilizadores, é algo que já faz parte do paradigma da sociedade; no entanto introduz vários desafios técnicos. Estes incluem a integração de diferentes formatos de média (conteúdo de alta definição, 3D, panorâmico, etc.), formas de construir diferentes narrativas, mecanismos de interação ou a possibilidade de aceder a conteúdo através de diferentes redes e dispositivos [1].

Por outro lado, a evolução da tecnologia e as expectativas dos utilizadores levam a que haja a oportunidade de dar um salto qualitativo na forma como estes conteúdos são tratados. Uma aposta a longo prazo seria a possibilidade de oferecer um ambiente imersivo, interativo e personalizável, que pudesse ter impacto significativo nas emoções dos utilizadores.

3.2 Solução proposta

A solução proposta para desenvolvimento teve como ponto de partida uma base assente em três pilares: ser possível visualizar e escutar conteúdo multimédia (vídeos, imagens, áudio, texto), haver interatividade e estar presente o carácter de inovação inerente a um projeto que pretende revolucionar a forma como o conteúdo multimédia é consumido.

Dessa forma, decidiu-se pela conceção e desenvolvimento de uma aplicação *web* com o principal foco nessas três premissas e com suporte para os principais navegadores *web*. Na Figura 10, apresenta-se o *mockup* da aplicação idealizada.

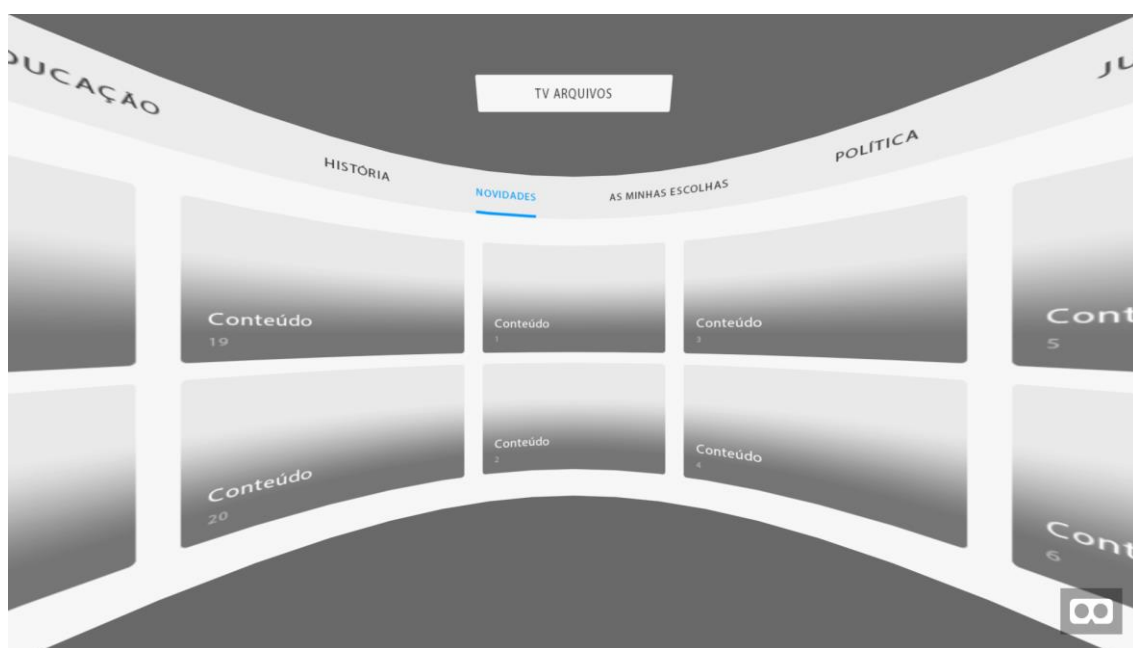


Figura 10 - *Mockup* concebido no início da dissertação.

3.2.1 Tecnologia

Dado o panorama descrito em 3.1 e tendo em mente o segundo e terceiro pilar, decidiu-se optar pelo desenvolvimento de uma aplicação com ambiente imersivo. Em tecnologia, frequentemente associa-se imersão a realidade virtual.

A realidade virtual é uma tecnologia informática que utiliza *headsets* próprios (ver Figura 11) para criar sensações que simulam a presença física dos utilizadores num ambiente virtual.

3.2.2 Ferramentas

Atualmente, os *headsets* de realidade virtual estão disponíveis para integração com o *smartphone*, para computador e para consolas de videojogos. Sendo uma aplicação concebida para navegadores *web*, a última categoria foi descartada.

Na figura seguinte apresenta-se os principais *headsets* disponíveis no mercado.

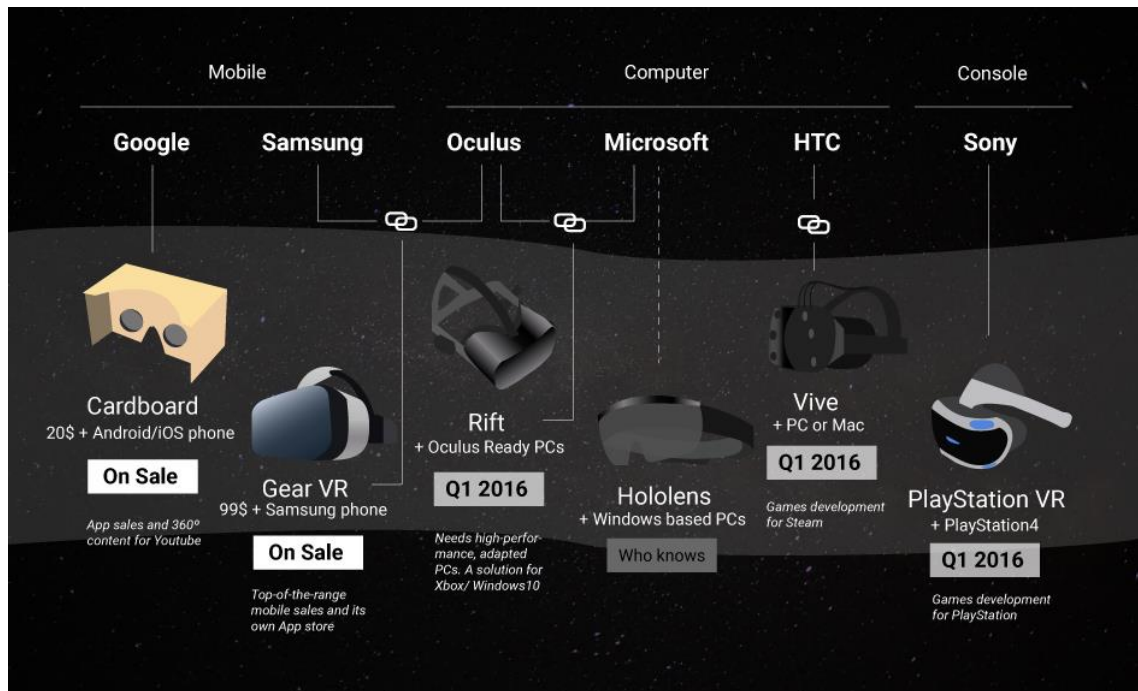


Figura 11 - Os principais *headsets* de realidade virtual disponíveis no mercado [9].

Para efeitos de testes da aplicação *web*, foi utilizado um *headset* similar ao Google Cardboard: VR Shinecon 3D VR Glasses Headset (ver Figura 12), do qual se salienta, das especificações existentes, a distância interpupilar ajustável e o campo de visão ajustável entre 85° e 90°.



Figura 12 - VR Shinecon 3D VR Glasses Headset [47].

Para o desenvolvimento da aplicação *web*, foram analisadas as ferramentas reunidas nas secções 6 a 9 do capítulo 2, que fossem de encontro à tecnologia e restantes características pretendidas. A escolha recaiu no A-Frame, pelo facto de este se basear e simplificar o desenvolvimento em WebGL e, ainda, por permitir a criação de cenas em realidade virtual na *web*. No capítulo seguinte dissecam-se melhor esta ferramenta.

3.3 Síntese

Neste capítulo apresentou-se a problemática desta dissertação e as decisões tomadas na conceção da solução proposta. Assim, explicitaram-se os três princípios em que a aplicação *web* se baseia, a escolha de uma tecnologia imersiva como a realidade virtual e o uso do A-Frame como ferramenta de desenvolvimento, juntamente com o *headset* escolhido para os respetivos testes.

Capítulo 4

A-Frame

Neste capítulo apresenta-se a ferramenta A-Frame usada para o desenvolvimento da aplicação *web*, tendo como objetivo permitir um melhor entendimento de como esta funciona, importante para a compreensão de algumas sessões do próximo capítulo.

Assim, primeiramente é explicado o que é o A-Frame, o seu modo de utilização e instalação; são apresentadas as principais ferramentas de auxílio e as suas principais características. Por último é feita uma referência à comunidade existente à volta do A-Frame.

4.1 O que é o A-Frame?

A-Frame é uma *web framework* para a construção de experiências de realidade virtual. Originalmente do Mozilla, o A-Frame foi desenvolvido para ser uma maneira fácil, mas poderosa, de desenvolver conteúdo em realidade virtual. Sendo um projeto de código aberto independente, o A-Frame tornou-se uma das maiores e mais acolhedoras comunidades de realidade virtual [30].

O A-Frame é baseado em HTML, o que o torna simples de usar. Apesar disso, não é apenas um criador de gráficos 3D, mas sim uma poderosa *framework* baseada em entidades e componentes que fornece uma estrutura declarativa, extensível e composta ao three.js [30].

Quanto à compatibilidade com os *headsets*, o A-Frame suporta a maioria, o que inclui o Vive, o Rift, o Daydream, o GearVR e o Cardboard, podendo até ser usado para realidade aumentada. Embora ofereça suporte para todo o espectro, o A-Frame visa definir experiências de realidade virtual interativas totalmente imersivas que vão além do conteúdo básico de 360°, fazendo uso integral da deteção posicional e controladores [30].

4.2 O básico

O A-Frame pode ser desenvolvido a partir de um ficheiro HTML sem precisar de instalar nada, uma vez que é constituído principalmente por HTML e JavaScript. O seu desenvolvimento pode ser feito através de editores de código, no navegador (um dos mais populares na comunidade do A-Frame será abordado na secção 4.2.2), ou localmente, criando um simples ficheiro HTML. Em ambos os casos, para utilizar o A-Frame, basta incluir no `<head>` uma `tag <script>` que aponta para a *build* CDN (ver Figura 13).

```
<head>
  <script src="https://aframe.io/releases/0.7.0/aframe.min.js"></script>
</head>
```

Figura 13 - A-Frame, CDN build (versão 0.7.0).

Em alternativa, pode-se fazer o *download* da *build* JS e importar o ficheiro localmente ou mesmo o *download* integral de uma versão do A-Frame.

Tendo já importado uma *build* JS, todo o código a desenvolver, relacionado com o A-Frame, deverá estar dentro das *tags* `<a-scene>` `</a-scene>`. Uma boa prática é utilizar o sistema de gestão de recursos, dado pela *tag* `<a-assets>`, que permite colocar os recursos num só lugar e fazer o seu pré-carregamento e armazenamento em *cache*, conseguindo-se assim um melhor desempenho.

De seguida apresenta-se um exemplo de uma cena básica, o chamado “*Hello, World!*” do A-Frame (Figura 14 e 15).

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Hello, WebVR! - A-Frame</title>
5    <meta name="description" content="Hello, WebVR! - A-Frame">
6    <script src="https://aframe.io/releases/0.7.0/aframe.min.js"></script>
7  </head>
8  <body>
9    <a-scene>
10     <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9" shadow></a-box>
11     <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E" shadow></a-sphere>
12     <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D" shadow></a-cylinder>
13     <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4" shadow></a-plane>
14     <a-sky color="#ECECEC"></a-sky>
15   </a-scene>
16 </body>
17 </html>
```

Figura 14 - Código do “*Hello, World!*” do A-Frame.

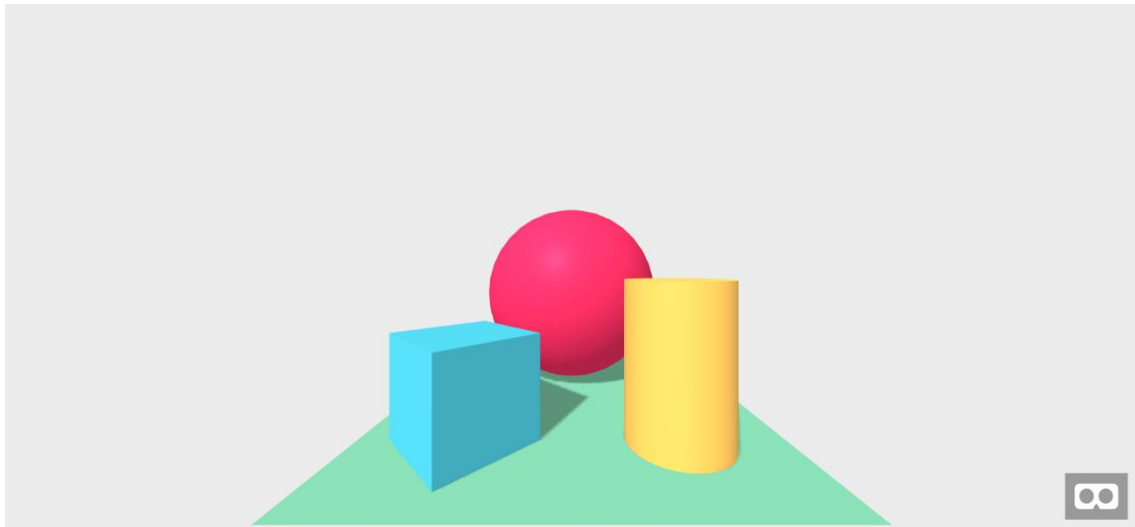


Figura 15 - Cenário do “Hello, World!” do A-Frame.

4.2.1 A-Frame School

O A-Frame School é um complemento à documentação existente que visa tornar mais fácil a adaptação do programador ao A-Frame. Este curso interativo de WebVR contém exercícios e lições visuais, passo a passo, que rapidamente cobrem os aspetos essenciais relativos ao A-Frame e às suas ferramentas auxiliares. Os conteúdos aqui abordados incluem:

- Glitch (o que é e como se utiliza) (ver 4.2.2);
- Consulta de páginas com exemplos de aplicações em A-Frame;
- Consulta e pequenas edições do “Hello, World!”;
- Adição de texturas;
- A-Frame Inspector (o que é e como se utiliza) (ver 4.2.3);
- As boas práticas do uso de componentes e entidades, ao invés de elementos primitivos (ver 4.3.4);
- A-Frame Registry (o que é e exercícios para aplicar) (ver 4.3.8);
- Uso do JavaScript;
- Componente cursor e a utilização de eventos;
- Adição de modelos 3D;
- Adição de controladores de mão e interatividade com objetos.

4.2.2 Glitch

O Glitch é um editor de código *online* que instantaneamente aloja e exibe páginas *web* (ver Figura 16). O editor suporta tanto código de *front-end* como *back-end*, assim como diversos ficheiros e diretórios. Com o Glitch é possível copiar projetos existentes e torná-los nossos, permitindo implementar mudanças, visualizar no momento cada mudança no código e partilhar facilmente para outros verem. Existem já vários projetos-exemplo feitos no Glitch e disponibilizados pelo A-Frame, que servem como base para o desenvolvimento de um novo projeto. Por estas razões, o Glitch é o editor de código *online* mais utilizado na comunidade A-Frame.

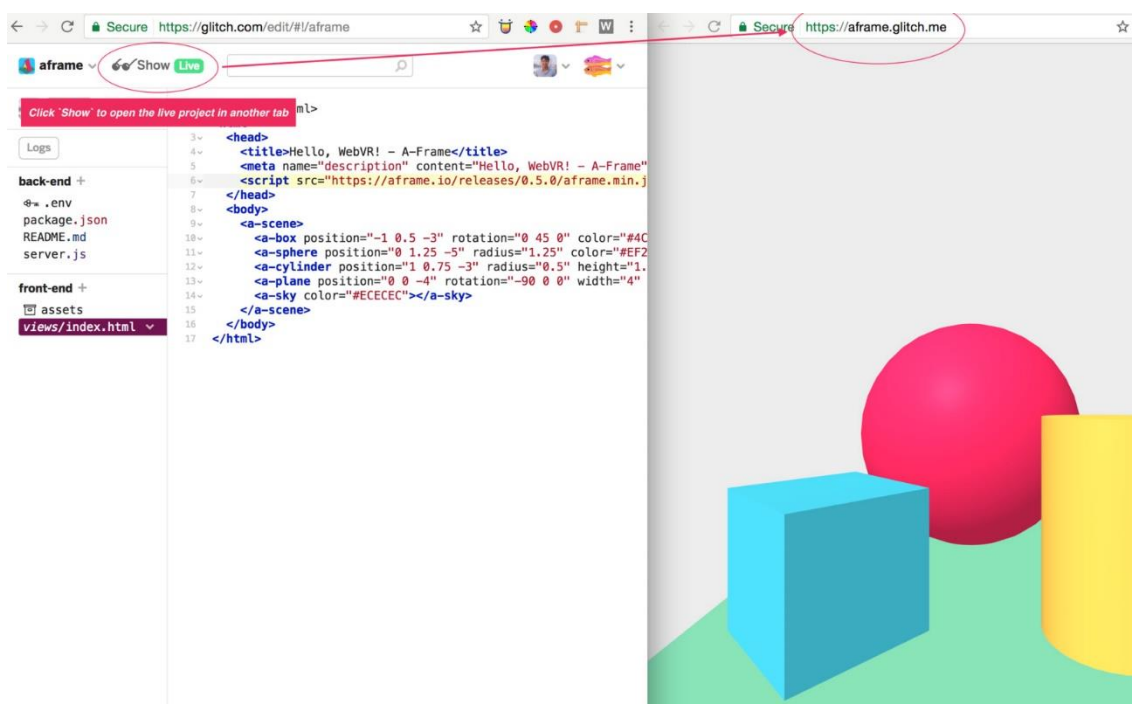


Figura 16 - Utilização do Glitch [30].

4.2.3 A-Frame Inspector

O A-Frame Inspector é uma ferramenta de desenvolvimento que permite obter uma visão diferente da cena e dos efeitos visuais das várias entidades ajustáveis. Pode ser visto como o análogo, em realidade virtual, ao inspetor dos navegadores. Para abrir o Inspector basta pressionar “<ctrl> + <alt> + i” em qualquer cena A-Frame. Depois, no painel do lado direito, é possível modificar as componentes ajustáveis (como por exemplo, atributos de geometria, etc.) e visualizar as alterações em tempo real (ver Figura 17).

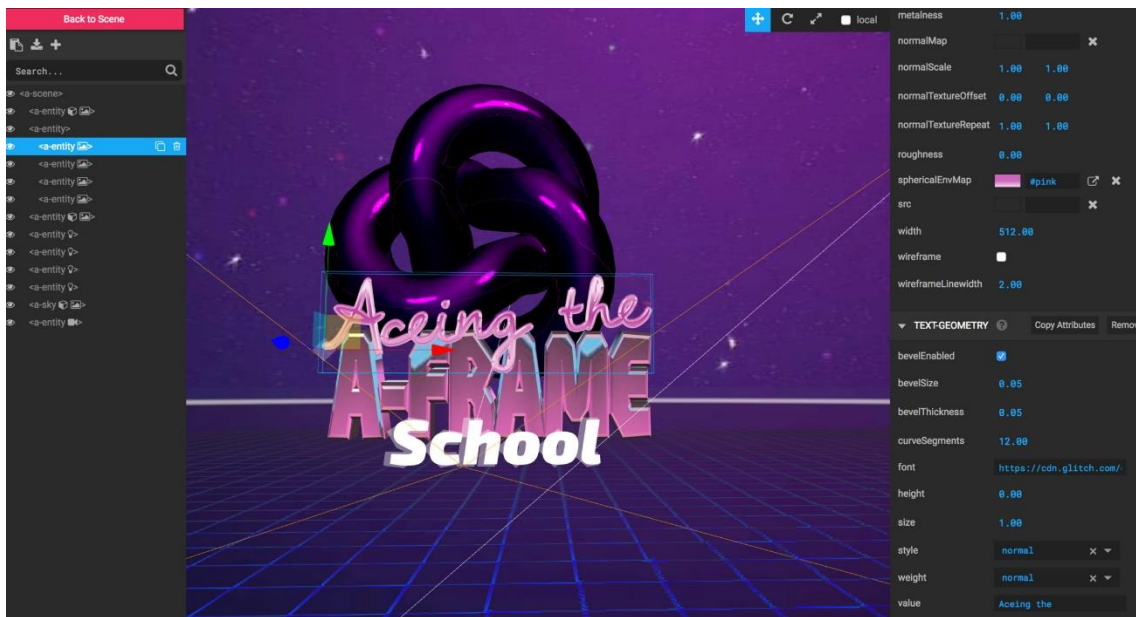


Figura 17 - Utilização do A-Frame Inspector [30].

4.2.4 A-Painter

O A-Painter é uma aplicação web da equipa do Mozilla VR, onde se pode pintar e desenhar em 3D. Para isso é necessário um *headset* e uns controladores de mãos, caso contrário, apenas é possível visualizar as criações de outros usando o rato e teclado ou o *smartphone* (Figura 18).

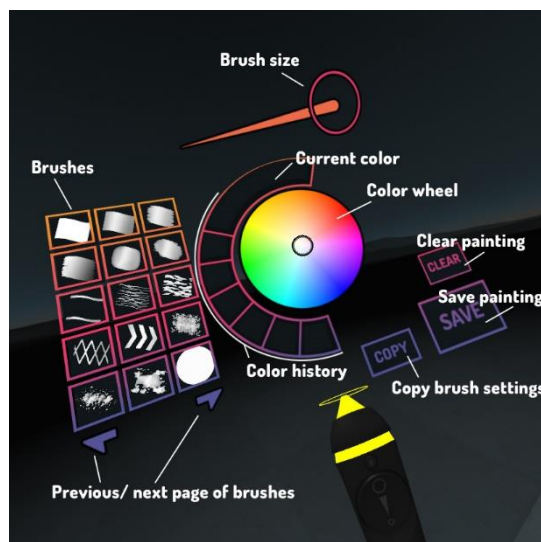


Figura 18 - A-Painter (menu principal do controlador) [48].

4.3 Características

Algumas das particularidades do A-Frame já ficaram patentes na secção anterior, mas de seguida apresentam-se, com um pouco mais de detalhe, as suas principais características [30].

4.3.1 Simplicidade

Apenas é preciso colocar uma *tag* `<script>` e uma *tag* `<a-scene>`. O A-Frame lida com a configuração 3D, com a configuração VR e com os controlos padrão. Não requer nenhuma instalação nem etapas de construção.

4.3.2 HTML declarativo e primitivas

O HTML é fácil de ler, entender e de copiar-colar. Sendo baseado em HTML, o A-Frame é acessível a todos: programadores *web*, entusiastas de realidade virtual, artistas, *designers*, educadores, criadores, crianças.

No A-Frame, são chamados de primitivas os elementos como `<a-box>`, `<a-sky>`, entre outros, que agrupam o padrão entidade-componente para ser mais apelativo para iniciantes.

4.3.3 Multiplataforma

As aplicações feitas no A-Frame são suportadas pelo Vive, Rift, Daydream, GearVR, Cardboard e respetivos controladores. Mesmo sem estes headsets, é possível utilizar o A-Frame num computador comum com um navegador *web* que suporte WebVR.

4.3.4 Sistema entidade-componente

O A-Frame é uma *framework* do three.js com um sistema de entidade-componente. O sistema entidade-componente é um padrão arquitetónico que permite grande flexibilidade ao definir entidades para os objetos de uma cena. Assim, o comportamento de uma entidade pode ser alterado durante a execução através da adição ou remoção de componentes. Isto elimina os problemas de ambiguidade das hierarquias de herança que são difíceis de entender, manter e estender. Além disso, promove um *design* limpo por meio de desacoplamento, encapsulamento, modularização e/ou reutilização, sendo a forma mais escalável de construir uma aplicação em VR em termos de complexidade.

4.3.5 Desempenho

O A-Frame é otimizado, desde a sua base, para WebVR, uma vez que os seus elementos não utilizam o mecanismo de layout do navegador *web*. As atualizações dos objetos 3D são feitas na memória com pouca sobrecarga sob uma única chamada “*requestAnimationFrame*”.

4.3.6 Compatibilidade

Uma vez que a *web* foi construída sobre HTML, o A-Frame é compatível com a maioria das bibliotecas, *frameworks* e ferramentas, incluindo React, Preact, Vue.js, d3.js, Ember.js, jQuery.

4.3.7 Componentes

Componentes da *framework* entidade-componente do A-Frame são módulos JavaScript que podem ser misturados, combinados e compostos em entidades para criar aparência, comportamento e funcionalidade. Regista-se o componente em JavaScript e utiliza-se declarativamente em HTML. Os componentes são configuráveis, reutilizáveis e compartilháveis. A maior parte do código de uma aplicação deve estar dentro de componentes.

Existem já várias componentes desenvolvidas e disponibilizadas por outros programadores na *web*.

4.3.8 A-Frame Registry

O A-Frame Registry é uma coleção de componentes publicados por outros programadores prontos a serem utilizados. Outro sítio, também bastante comum entre a comunidade A-Frame, para a procura de componentes é o npm (Noxious Plant Miasma, www.npmjs.com), pesquisando por “*keywords:afree*”.

4.4 Comunidade

O A-Frame possui três elementos na sua equipa núcleo ao qual se juntam mais de 200 contribuidores. A maior parte da discussão de desenvolvimento acontece no repositório do projeto no GitHub e qualquer pessoa pode estar a par do estado atual de cada versão, bem como de problemas conhecidos.

Na página principal do A-Frame encontra-se uma ligação para o *blog* oficial onde, todas as semanas, no “*Week of A-Frame*”, são recolhidas as últimas notícias, eventos futuros e projetos em desenvolvimento por outras pessoas.

O Slack do A-Frame (<https://aframevr.slack.com>) é o sítio ideal para trocar ideias com outros programadores. Os canais disponíveis neste momento são: *#general*, *#questions*, *#issues*, *#projects* e *#random*. No entanto, para colocar dúvidas ou obter ajuda técnica, o A-Frame recomenda a colocação das mesmas no Stack Overflow.

Por fim, o A-Frame possui conta no Twitter (@aframevr), grupo no Facebook (facebook.com/groups/aframevr) e subreddit (/r/WebVR/).

4.5 Síntese

Neste capítulo caracterizou-se a ferramenta utilizada para o desenvolvimento da aplicação *web*, o A-Frame, que é uma *web framework* para a construção de experiências de realidade virtual.

Explicaram-se as diversas possibilidades que existem para se começar a utilizar o A-Frame, sendo que a forma mais fácil é a de usar um editor de código no navegador *web* e incluir uma *tag* `<script>` que aponte para a versão da *build* JS pretendida, juntamente com uma *tag* `<a-scene>` onde se coloca o código a desenvolver relacionado com o A-Frame.

Apuraram-se outros projetos relacionados com o A-Frame: A-Frame School, A-Frame Inspector, A-Painter e A-Frame Registry.

Concluiu-se, também, que o A-Frame possui uma vasta comunidade, com diversos canais de comunicação, sendo o GitHub, o Slack e o Stack Overflow os meios ideais para estar a par dos desenvolvimentos e problemas conhecidos do A-Frame, trocar ideias com outros programadores e colocar dúvidas ou obter ajuda técnica, respetivamente.

Capítulo 5

A aplicação: MondoVR

Neste capítulo apresenta-se a aplicação *web* desenvolvida, MondoVR, com a finalidade de mostrar e analisar cada elemento que a integra.

Para tal, começa-se por fazer uma introdução ao MondoVR e apresenta-se a sua arquitetura. Seguidamente, explica-se o *design*, a organização e cada componente e entidade que a incorpora. Por último, aborda-se o funcionamento da aplicação, os problemas conhecidos e discute-se os resultados.

5.1 Introdução ao MondoVR

MondoVR é uma aplicação *web*, feita em A-Frame, que permite a visualização de conteúdos multimédia em ambiente de realidade virtual. Desenvolvida a pensar, primariamente, numa utilização em conjunto com um *headset* de realidade virtual, esta *web app* pode ser, também, utilizada no computador através de um navegador *web*.

O objetivo desta aplicação é permitir uma visualização simples, intuitiva e ao mesmo tempo fornecer uma componente de imersão no consumo de conteúdo multimédia.

O MondoVR encontra-se disponível nas seguintes hiperligações:

- “<https://paginas.fe.up.pt/~ee08270/proj/app/>”;
- “<http://vrmondoapp.ga>” (domínio temporário).

5.2 Arquitetura

A aplicação *web* possui uma arquitetura simples do tipo cliente/servidor, focada essencialmente nas camadas superiores, ou seja, na interface do utilizador, no lado do cliente e na camada de apresentação, no lado do servidor.

Assim, quando o utilizador pretende abrir a aplicação, o seu navegador *web* efetua um pedido HTML ao servidor, que envia um ficheiro HTML. De seguida, esse ficheiro informa o navegador que precisa de mais ficheiros (js, imagens, etc.) e o servidor envia os respetivos ficheiros. Posteriormente, o JavaScript do cliente monta tudo no seu navegador *web* (ver Figura 19). Quando é acionado um clique para reproduzir um vídeo, o JavaScript do cliente faz um pedido a requisitar o respetivo vídeo, que é enviado pelo servidor, e depois atualiza a entidade na página HTML (ver Figura 20).

A aplicação encontra-se alojada na área pessoal de armazenamento de dados em servidores da FEUP.

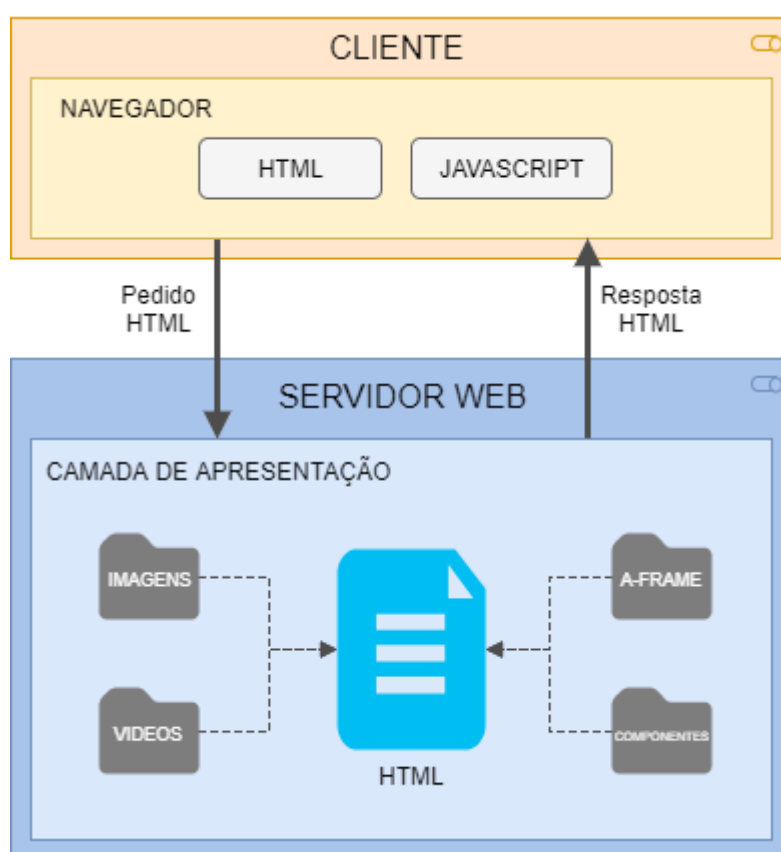


Figura 19 - Diagrama da arquitetura da aplicação.

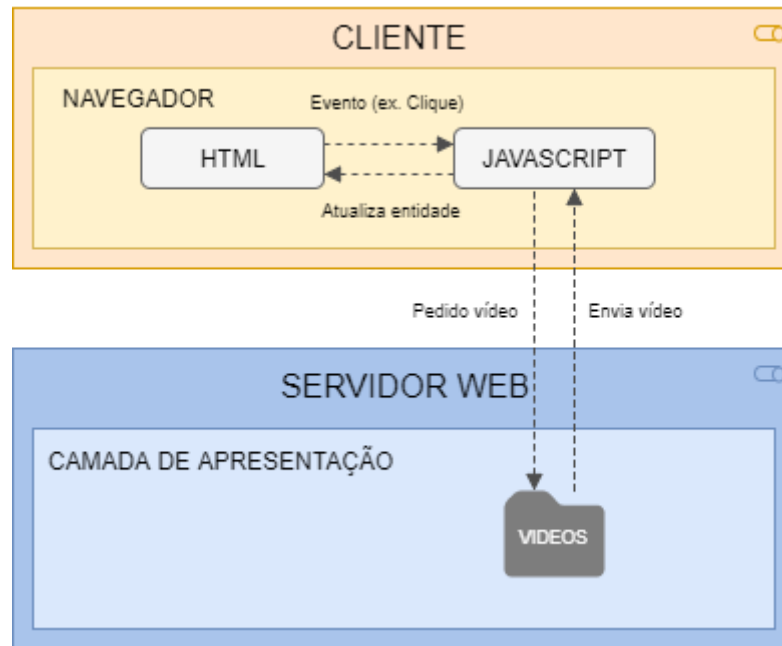


Figura 20 - Diagrama sobre o funcionamento da aplicação na visualização de um vídeo.

5.3 Web design

Em realidade virtual ainda não há regras de *design* claras, devido às diversas técnicas de interação que existem. A preocupação fundamental do *web designer* consiste em agregar os conceitos de usabilidade com a arquitetura de informação de forma a garantir que o utilizador consiga atingir o seu objetivo de um modo agradável e intuitivo.

Durante a conceção da aplicação, decidiu-se que seria interessante explorar as potencialidades que a realidade virtual oferece e desenvolver uma estrutura em 360°. Para esse efeito, optou-se por um *layout* circular, no plano “xz”, circundando o utilizador, que se encontra no seu centro (ver Figura 21).

Assim, nas subsecções seguintes, abordam-se os principais elementos da estrutura do *web design*, bem como a sua disposição.

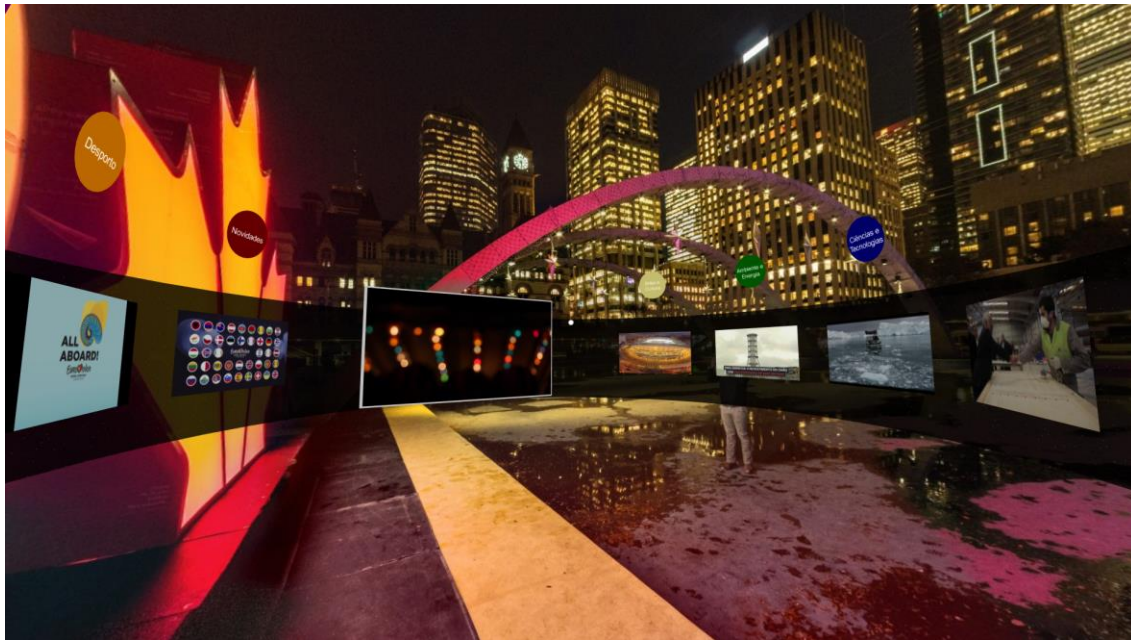


Figura 21 - Plano afastado da aplicação MondoVR.

5.3.1 Organização

A disposição dos elementos que integram a estrutura de uma aplicação para realidade virtual, nomeadamente o seu posicionamento, tem um grande impacto na experiência do utilizador. Desta forma, na presente aplicação, os elementos em maior destaque são os seguintes:

- Miniaturas de vídeos distribuídas seguindo um *layout* circular no plano “xz”;
- Uma imagem curva, designada pela primitiva *<a-curvedimage>*, que serve como fundo às miniaturas dos vídeos;
- Círculos, relativos às categorias, distribuídos seguindo um *layout* circular no plano “xz”;
- Um plano definido para reprodução dos vídeos;
- Um plano definido para a amostragem das informações relativas ao vídeo em exibição.

Na Figura 22, podemos verificar que podem existir até nove miniaturas de vídeos dispostas a 8 metros de distância da posição do utilizador/câmara, que está em (0,0,0), e a 30 graus de distância entre si. Caso o número de miniaturas a apresentar seja inferior, serão ocupadas as posições por ordem crescente. A razão dessa escolha deve-se ao facto de, quando estamos a visualizar uma miniatura, a miniatura vizinha se encontrar no campo de visão e captar a atenção do utilizador. Para além disso, esta disposição permite uma organização temporal, neste caso, dos vídeos mais recentes para os mais antigos.

As categorias, não representadas na figura acima, estão posicionadas no mesmo sítio bidimensional que as miniaturas das posições 1, 2, 3, 7, 8, 9, mas com a posição do eixo do “y” em 3 metros. A categoria correspondente à posição 9 é referente às novidades, isto é, aos nove vídeos disponíveis mais recentes, sendo também a categoria apresentada na abertura da aplicação.

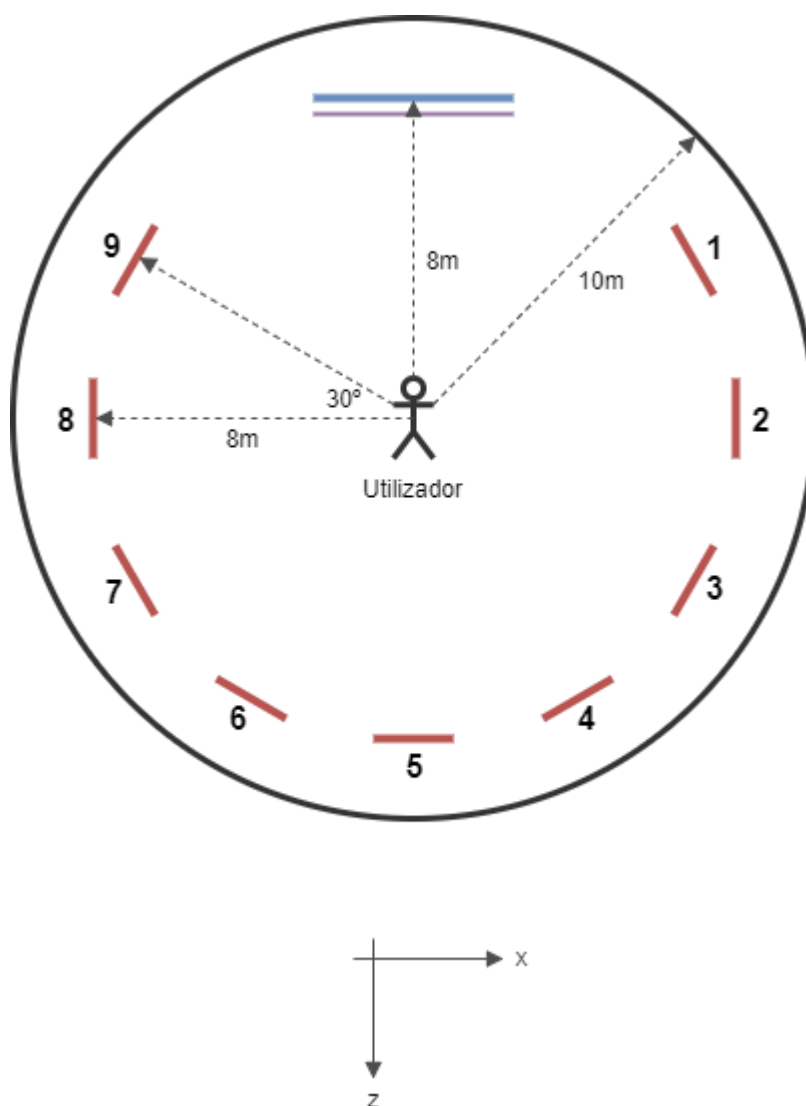


Figura 22 - Organização em 2D dos principais elementos da aplicação.

Relativamente aos restantes elementos, a imagem curva, o plano para a reprodução dos vídeos e o plano das informações sobre os vídeos, posicionam-se respetivamente a 10, 8 e 7,5 metros da câmara. Este último tem uma posição do eixo do “y” de -4,5 metros.

Na secção 5.6, serão analisados todos os atributos não só destes, mas também de outros elementos da aplicação.

5.3.2 O ecrã

O ecrã, representado pelo plano onde são reproduzidos os vídeos, é o elemento com maior foco de toda a aplicação. Aquando da abertura da aplicação, um vídeo-exemplo é reproduzido automaticamente e repetidamente até ser dado um clique na miniatura de um vídeo.

O motivo pelo qual se optou por visualizar os vídeos num ecrã único, ao invés da sua visualização, numa escala maior, nas posições das suas miniaturas, foi para possibilitar um maior conforto aos utilizadores com *headset* de realidade virtual. Para estes, ao rodarem a cabeça para seleccionar um vídeo, seria incómodo serem obrigados a manterem essa posição durante toda a duração do mesmo. Deste modo, chegou-se à conclusão que seria vantajoso visualizar os vídeos no campo de visão inicial.

Relativamente ao tamanho do ecrã, uma vez mais deu-se maior foco aos utilizadores que optem pelo modo realidade virtual, pois neste o ecrã ocupa a maior parte do campo de visão. Já para os utilizadores que utilizem o modo *fullscreen* num computador, o ecrã poderá ser algo pequeno.

5.3.3 O cursor

O cursor é representado por um anel, inicialmente branco, colocado 3 metros em frente da câmara, estando preso ao olhar do utilizador (no modo realidade virtual). Este anel varia o seu comportamento dependendo se está por cima de um objeto clicável ou não. Caso não esteja, o anel apresenta um tamanho muito pequeno, parecendo quase um ponto. No caso de estar, o anel aumenta os raios interno e externo, e durante 3 segundos vai mudando a sua cor para vermelho até ser desencadeado o clique, no final desses 3 segundos, voltando ao estado anterior (ver Figura 23).



Figura 23 - Os diferentes estados do cursor: à esquerda, o cursor encontra-se sobre um objeto não clicável; ao centro, o cursor encontra-se sobre um objeto clicável; e à direita, o cursor encontra-se no estado iminente que antecede o clique.

5.4 Conteúdos

Foram reunidos 15 vídeos de alta definição e 15 imagens, correspondentes às miniaturas dos vídeos. Estes estão divididos em cinco categorias, cada uma com três vídeos e três imagens.

Para cada vídeo foram recolhidas as seguintes informações:

- Título do vídeo;
- Duração do vídeo;
- Descrição do vídeo;
- Data do vídeo;
- Imagem da miniatura do vídeo.

A cada vídeo foi atribuído:

- Um ID, seguindo o formato video##;
- Uma categoria.

As categorias escolhidas foram:

- Artes e cultura;
- Ambiente e energia;
- Ciências e tecnologias;
- Desporto;
- Política.

Os conteúdos disponíveis são utilizados meramente para fins demonstrativos do funcionamento da aplicação e encontram-se localmente nos diretórios:

- /images/thumbnails/
- /videos/

5.5 Componentes

Os componentes assumem um papel fundamental no desenvolvimento de uma aplicação em A-Frame. A maior parte da aparência, do comportamento e da funcionalidade são feitos em módulos JavaScript, que, depois de registado o componente, podem ser configuráveis, reutilizáveis e compartilháveis.

Assim, nesta aplicação, foram desenvolvidos quatro componentes e utilizados outros três, disponibilizados *online* no A-Frame Registry e no npm (Noxious Plant Miasma), e desenvolvidos pelo co-criador do A-Frame, o qual é conhecido na comunidade pela alcunha *ngokeyin*.

Nas subsecções seguintes apresenta-se cada um dos componentes.

5.5.1 *aframe-template-component*

O componente *template* utiliza-se essencialmente para encapsular grupos de entidades, passar dados para gerar entidades ou para iteração. Evita-se assim o uso excessivo do copiar-colar, criando-se uma entidade modelo. Uma maneira de definir um modelo é através de uma *tag de script* em `<a-assets>` (ver Figura 24).

```
<a-scene>
  <a-assets>
    <script id="boxesTemplate">
      <a-box color="{box1color}" position="-1 0 -5"></a-box>
      <a-box color="{box2color}" position="0 1 -5"></a-box>
      <a-box color="{box3color}" position="1 0 -5"></a-box>
    </script>
  </a-assets>

  <a-entity template="src: #boxesTemplate"
    data-box1color="red" data-box2color="green" data-box3color="blue"></a-entity>
</a-scene>
```

Figura 24 - Exemplo da utilização do componente *template*.

Desta forma, na aplicação, este componente foi particularmente útil na criação dos planos que contêm as miniaturas dos vídeos. Definiu-se no modelo: a geometria (incluindo as dimensões); eventos de interação com o cursor (como por exemplo, a visibilidade da legenda e das informações do vídeo respetivo); e o som de clique. Depois, para cada plano, foram atribuídos individualmente a fonte do material (imagem da miniatura do vídeo), o vídeo, a legenda e as respetivas informações (ver Figura 25).


```

<a-scene>
  <a-assets>
    <script id="plane" type="text/html">
      <a-entity set-video class="clickable"
        geometry="primitive: plane; height: 1.8; width: 3.2"
        material="shader: standard; src: ${thumb}"
        event-set__1="_event: mouseleave; _target: ${txt}; visible: false"
        event-set__2="_event: mouseenter; _target: ${txt}; visible: true"
        event-set__3="_event: click; _target: #screen; material.src: ${vid}"
        event-set__4="_event: click; _target: #d-title; visible: true"
        event-set__5="_event: click; _target: #d-description; visible: true"
        event-set__6="_event: click; _target: #d-duration; visible: true"
        event-set__7="_event: click; _target: #d-cat; visible: true"
        event-set__8="_event: click; _target: #d-date; visible: true"
        event-set__9="_event: click; _target: #detailsplanebg; visible: true"
        event-set__10="_event: click; _target: #detailsplane; visible: true"
        sound="on: click; src: #click-sound">
      </a-entity>
    </script>
  </a-assets>

  <a-entity id="content01" rotation="0 -60 0" template="src: #plane"
    data-thumb="#thumb01" data-txt="#txt01" data-vid="#video01"></a-entity>
</a-scene>

```

Figura 25 - Excerto do código da aplicação que representa a utilização da componente *template*.

5.5.2 *aframe-layout-component*

A grande vantagem do componente *layout* consiste em não ser necessário posicionar individualmente cada entidade que se queira que siga uma determinada disposição. Para isso, cria-se uma entidade “pai”, que envolve todas as entidades “filho”, e anexa-se o componente *layout*, que posiciona automaticamente as entidades “filho” no *layout* desejado (ver Figura 26).

```

<a-entity layout="type: line; margin: 2">
  <a-box position="0 0 0"></a-box>
  <a-box position="2 0 0"></a-box>
  <a-box position="4 0 0"></a-box>
</a-entity>

```

Figura 26 - Exemplo da utilização do componente *layout*.

Na aplicação, este componente foi utilizado para posicionar os planos das miniaturas dos vídeos e as suas legendas. Definiu-se o tipo de *layout* “círculo”, com raio de 8 metros, ângulo de 30 graus e o plano “xz” (ver Figura 27).

```

<a-entity layout="type: circle; radius: 8; angle: 30; plane: xz">
  <a-entity id="content01" rotation="0 -60 0" template="src: #plane"
    data-thumb="#thumb01" data-txt="#txt01" data-vid="#video01"></a-entity>
  <a-entity id="content02" rotation="0 -90 0" template="src: #plane"
    data-thumb="#thumb02" data-txt="#txt02" data-vid="#video02"></a-entity>
  <a-entity id="content03" rotation="0 -120 0" template="src: #plane"
    data-thumb="#thumb03" data-txt="#txt03" data-vid="#video03"></a-entity>
</a-entity>

<a-entity layout="type: circle; radius: 8; angle: 30; plane: xz" position="0 -1.2 0">
  <a-entity id="txt01" rotation="0 -60 0" mixin="thumbs-title-format title01" visible="false"></a-entity>
  <a-entity id="txt02" rotation="0 -90 0" mixin="thumbs-title-format title02" visible="false"></a-entity>
  <a-entity id="txt03" rotation="0 -120 0" mixin="thumbs-title-format title03" visible="false"></a-entity>
</a-entity>

```

Figura 27 - Excerto do código da aplicação que representa a utilização da componente *layout*.

5.5.3 *aframe-event-set-component*

A função do componente *event-set* é a de adicionar uma resposta visual na sequência de um evento, como, por exemplo, um clique ou uma passagem do cursor por cima de um objeto. Essa resposta visual poderá ser uma mudança de cor, alteração de posição ou rotação, entre outras. Os atributos a definir são: o evento, a entidade alvo (opcional) e os atributos do objeto a modificar (ver Figura 28).

```

<a-scene>
  <a-box color="green"
    event-set__click="material.color: red; scale: 2 2 2"
    event-set__mouseenter="material.color: blue"></a-box>
</a-scene>

```

Figura 28 - Exemplo da utilização do componente *event-set*.

Este componente foi empregue nas entidades relativas às categorias, para adicionar uma alteração visual de escala à passagem do cursor, e nos planos das miniaturas dos vídeos, para controlar a visibilidade da legenda e informações sobre o vídeo.

Na Figura 25 é possível observar a utilização deste componente na aplicação desenvolvida.

5.5.4 *set-bg.js*

Este componente, registado com o nome de *set-bg*, é utilizado na entidade correspondente ao ecrã e foi desenvolvido para realizar as seguintes tarefas:

- Quando o cursor entra na área da entidade ecrã:
 - a. Reduz a opacidade da imagem de fundo da aplicação para 10%;
 - b. Reproduz o vídeo inicial ou, caso já tenha sido selecionado um vídeo, reproduz o mesmo.

- Quando o cursor sai da área da entidade ecrã:
 - a. Repõe a opacidade da imagem de fundo da aplicação para 100%;
 - b. Coloca em pausa o vídeo que se encontra em reprodução.

5.5.5 *set-video.js*

Este componente, registado com o nome de *set-video*, é utilizado na entidade modelo dos planos referentes às miniaturas dos vídeos (ver Figura 25) e foi desenvolvido para realizar as seguintes tarefas, aquando de um clique nas entidades que advêm desse modelo:

- Parar a reprodução do vídeo atual;
- Determinar qual o vídeo clicado;
- Reproduzir o vídeo clicado;
- Determinar as informações sobre o vídeo clicado;
- Apresentar as informações sobre o vídeo clicado;
- Calcular o tempo decorrido do vídeo em reprodução.

5.5.6 *set-category.js*

Este componente, registado com o nome de *set-category*, é utilizado nas entidades correspondentes às categorias e foi desenvolvido para realizar as seguintes tarefas, aquando de um clique nas mesmas entidades:

- Caso a categoria não seja “Novidades”:
 - a. Apresentar as miniaturas dos vídeos da categoria selecionada;
 - b. Apresentar as legendas dos vídeos da categoria selecionada;
 - c. Atribuir os vídeos respetivos da categoria selecionada às miniaturas;
 - d. Remover as posições não utilizadas.
- Caso a categoria seja “Novidades”:
 - a. Apresentar as miniaturas dos nove vídeos atribuídos inicialmente;
 - b. Apresentar as legendas dos nove vídeos atribuídos inicialmente;
 - c. Atribuir os nove vídeos designados inicialmente às miniaturas.

5.5.7 *set-randombg.js*

Este componente, registado com o nome de *set-randombg*, é utilizado na primitiva *<a-sky>*, que corresponde à imagem de fundo, e foi desenvolvido para realizar a seguinte tarefa:

- Alternar as imagens de fundo a cada 15 segundos com um efeito de desvanecimento.

5.6 Entidades e primitivas

Nesta secção dissecam-se as entidades e primitivas utilizadas, e faz-se uma análise às suas propriedades e atributos.

5.6.1 *<a-sky>*

A primitiva *sky* adiciona uma cor de fundo ou uma imagem de 360 graus a uma cena A-Frame. Uma *sky* é uma esfera de grandes dimensões com uma cor ou textura mapeada para dentro.

Na Figura 29 estão representados os atributos relacionados com esta primitiva.

Attribute	Component Mapping	Default Value
color	material.color	#FFF
metalness	material.metalness	0
opacity	material.opacity	1
phi-length	geometry.phiLength	360
phi-start	geometry.phiStart	0
radius	geometry.radius	5000
repeat	material.repeat	None
roughness	material.roughness	0.5
segments-height	geometry.segmentsHeight	20
segments-width	geometry.segmentsWidth	64
shader	material.shader	flat
side	material.side	front
src	material.src	None
theta-length	geometry.thetaLength	180
theta-start	geometry.thetaStart	0
transparent	material.transparent	false

Figura 29 - Atributos da primitiva *<a-sky>* [49].

Para a aplicação, utilizou-se apenas o atributo “*src*”, para adicionar uma imagem equiretangular, e juntou-se a componente *set-randombg* para efetuar a alteração de imagem a cada 15 segundos.

5.6.2 <a-curvedimage>

A primitiva *curvedimage* permite criar imagens que se dobras ao redor do utilizador. A sua utilização é sobretudo útil para elementos de grandes dimensões, pois cada pixel fica à mesma distância do utilizador. Na prática, uma *curvedimage* é um cilindro aberto, com texturas mapeadas para dentro do cilindro.

Na Figura 30 estão representados os atributos relacionados com esta primitiva.

Attribute	Component Mapping	Default Value
color	material.color	#FFF
height	geometry.height	1
metalness	material.metalness	0
opacity	material.opacity	1
open-ended	geometry.openEnded	true
radius	geometry.radius	2
repeat	material.repeat	None
roughness	material.roughness	0.5
segments-height	geometry.segmentsHeight	18
segments-radial	geometry.segmentsRadial	48
shader	material.shader	flat
side	material.side	double
src	material.src	None
theta-length	geometry.thetaLength	270
theta-start	geometry.thetaStart	0
transparent	material.transparent	true

Figura 30 - Atributos da primitiva <a-curvedimage> [50].

Assim, nesta aplicação, utilizaram-se os seguintes atributos:

- *src* - para adicionar uma textura;
- *height* - 4 metros de altura;
- *radius* - 10 metros de raio;
- *theta-length* - 360 graus de extensão;
- *repeat* - textura repetida 15 vezes;
- *opacity* - opacidade igual a 0.8, ou seja, 80%.

5.6.3 Entidades relacionadas com as miniaturas dos vídeos e legendas

Tal como foi dito em 5.5.1, para a criação dos planos das miniaturas dos vídeos, desenvolveu-se uma entidade modelo (ver Figura 25). Nesta entidade, para além do componente *set-video*, especificaram-se os componentes *geometry* e *material*, nativos do A-Frame. Relativamente ao *geometry*, das diversas geometrias possíveis, escolheu-se *plane*, uma superfície plana, e especificou-se *height* (altura) de 1,8 metros e *width* (largura) de 3,2 metros, mantendo-se assim a proporção 16:9. Quanto ao *material*, foi decidido que a *src* seria variável para cada *plane*. Ainda na entidade modelo, definiram-se vários eventos, nomeadamente:

- Quando o cursor estiver por cima de um *plane*, tornar a sua legenda visível.
- Caso o cursor não esteja por cima de um *plane*, tornar a sua legenda invisível.
- Quando acionado um clique num *plane*, carregar no ecrã o vídeo atribuído.
- Sempre que se efetue um clique num *plane*, tornar visíveis as informações sobre o vídeo e o respetivo plano onde estão assentes.
- Sempre que houver um clique num *plane*, reproduzir um som de clique.

Seguidamente, criou-se uma entidade para cada *plane*, baseada na entidade modelo. Como se pode ver na Figura 27, definiu-se que estas entidades estariam dispostas segundo um *layout* circular, no plano “xz”, de raio igual a 8 metros e distanciadas 30 graus entre si. A cada entidade atribuiu-se a rotação, a entidade modelo, a *src* do *material*, a referência à sua legenda e a referência ao seu vídeo. Ainda na Figura 27 estão presentes, igualmente, as entidades das legendas, que seguem a mesma disposição dos *planes*, mas posicionadas 1,2 metros abaixo. Na Figura 31 pode-se ver o resultado visual da entidade de um *plane* e da entidade da respetiva legenda.

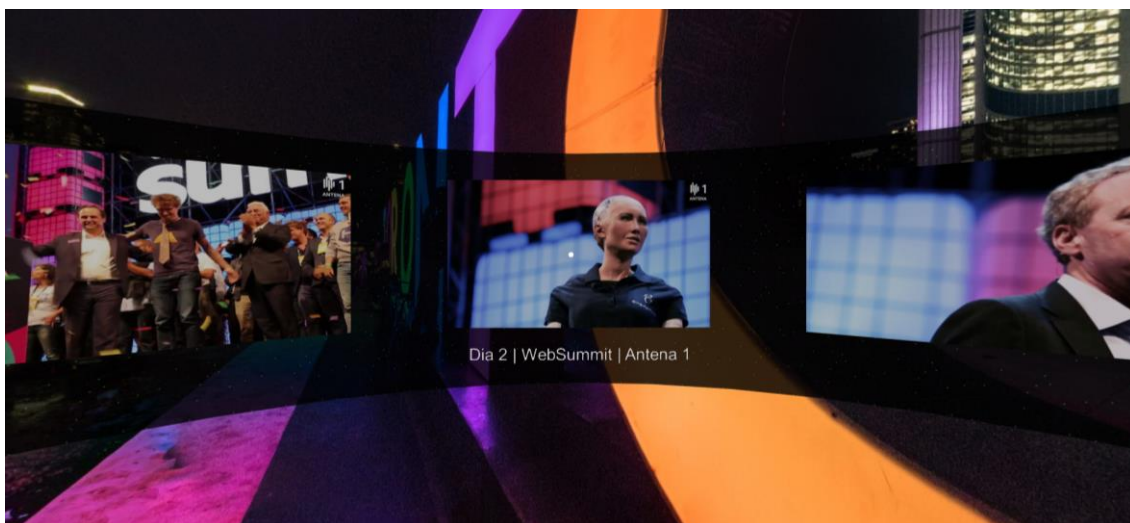


Figura 31 - Resultado visual de uma entidade *plane* e legenda respetiva.

5.6.4 Entidades relacionadas com o ecrã

Para criar o ecrã, desenvolveu-se uma entidade “pai” e duas entidades “filho”. A entidade “pai” serve exclusivamente para posicionar espacialmente as entidades “filho”. As outras duas entidades destinam-se ao plano para a reprodução dos vídeos e a um segundo plano para a produção de um efeito de moldura sobre o primeiro. Na Figura 32 apresenta-se o código relativo a estas entidades.

```
<a-entity id="videoscreen" position="0 0.25 -8">
  <a-entity id="screenbg"
    geometry="primitive: plane; width: 8.2; height: 4.65"
    material="color: darkgrey" position="0 0 -0.1"></a-entity>
  <a-entity id="screen" set-bg class="clickable"
    geometry="primitive: plane; width: 8; height: 4.5"
    material="src: #disco; shader: flat"></a-entity>
</a-entity>
```

Figura 32 - Código das entidades relacionadas com o ecrã.

Analisando a figura anterior, verifica-se que estas entidades estão afastadas 8 metros da câmara no eixo “z”. A entidade que serve de ecrã tem atribuído o componente *set-bg* e possui geometria do tipo *plane*, largura de 8 metros, altura de 4,5 metros e a fonte do material endereçada ao vídeo-exemplo. Já a entidade que serve de moldura também possui geometria do tipo *plane*, posicionada 10 centímetros atrás do ecrã, com dimensões ligeiramente maiores (8,2 metros de largura; 4,65 metros de altura) e tem material de cor cinzenta escura. Na Figura 33 pode ser visualizado o resultado.

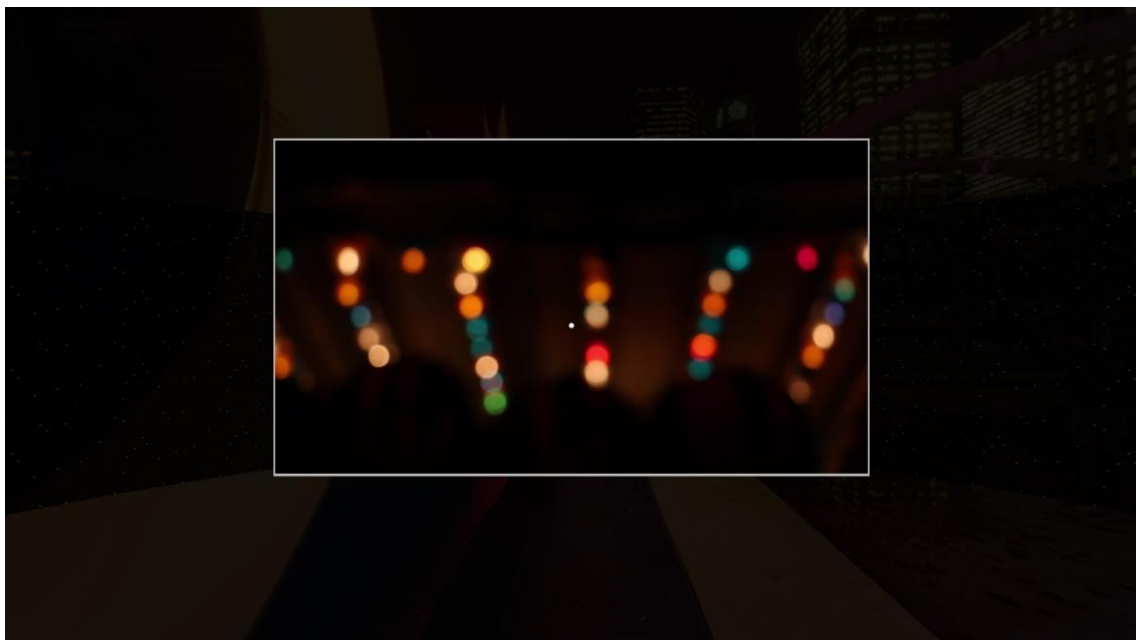


Figura 33 - Resultado visual das entidades relacionadas com o ecrã.

5.6.5 Entidades e primitivas relacionadas com as informações sobre os vídeos

A superfície que contém as informações sobre os vídeos foi criada de forma similar à do ecrã. Utilizaram-se duas primitivas “filho”, uma para o plano onde assentam as informações e outra para a produção do efeito moldura, e uma entidade “pai”, que posiciona espacialmente as entidades anteriores.

Foram também adicionadas mais cinco entidades para posicionar os campos “título”, “descrição”, “duração”, “categoria” e “data”.

Na Figura 34 é explicitado o código respetivo a estas entidades.

```
<a-entity id="details" position="0 -4.5 -7.5" rotation="-20 0 0">
  <a-plane id="detailsplanebg" color="darkgrey" position="0 -0.025 -0.1" width="8.35" height="3.1" visible="false"></a-plane>
  <a-plane id="detailsplane" color="black" width="8.2" height="3" visible="false">
    <a-entity id="d-title" position="-0.9 1.3 0.2" text="value:..." visible="false"></a-entity>
    <a-entity id="d-description" position="0 0.5 0.2" text="value:..." visible="false"></a-entity>
    <a-entity id="d-duration" position="-0.9 -1.3 0.2" text="value:..." visible="false"></a-entity>
    <a-entity id="d-cat" position="0.9 -1.3 0.2" text="value:..." visible="false"></a-entity>
    <a-entity id="d-date" position="0.9 1.3 0.2" text="value:..." visible="false"></a-entity>
  </a-plane>
</a-entity>
```

Figura 34 - Código das entidades e primitivas relacionadas com as informações sobre os vídeos.

Pode verificar-se que estas entidades e primitivas estão posicionadas a -7,5 metros no eixo do “z” e a -4,5 metros no eixo do “y”, com uma rotação de -20 graus sobre o eixo do “x”. A primitiva que corresponde ao plano onde assentam as informações dos vídeos possui os atributos: cor preto; largura de 8,2 metros; altura de 3 metros. Já a primitiva que corresponde à moldura, encontra-se posicionada ligeiramente atrás da primitiva anterior e possui os atributos: cor cinzento escuro; largura de 8,35 metros; altura de 3,1 metros (ver Figura 35).

As entidades relativas aos campos “título”, “descrição”, “duração”, “categoria” e “data” estão posicionadas dentro da primeira primitiva aqui referida e possuem diversas propriedades configuráveis, tais como, alinhamento horizontal e vertical, tamanho de letra, entre outros.

Todas estas entidades e primitivas encontram-se com a propriedade visibilidade em falso, pois esta só se quer visível quando acontecer um clique para a reprodução de um vídeo.



Figura 35 - Resultado visual das entidades e primitivas relacionadas com as informações sobre os vídeos.

5.6.6 Entidades relacionadas com as categorias

Para criar as categorias, desenvolveram-se seis entidades correspondentes às cinco categorias dos conteúdos mais a categoria “Novidades”. Utilizou-se um elemento `<a-mixin>`, colocado em `<a-assets>`, que permite agrupar conjuntos de propriedades mais utilizadas, de forma a facilitar a sua reutilização (ver Figura 36).

```
<a-assets>
...
  <a-mixin id="category" geometry="primitive: circle; radius: 1"
    text="..." scale="0.6 0.6 1" sound="on: click; src: #click-sound"></a-mixin>
...
</a-assets>

<a-entity id="cat2esq" set-category class="clickable" mixin="category"
  material="color: darkorange"
  position="-8 3 0" rotation="10 90 0" text="value: Desporto"
  event-set__1="_event: mouseenter; scale: 0.8 0.8 1"
  event-set__2="_event: mouseleave; scale: 0.6 0.6 1">
</a-entity>
```

Figura 36 - Código da entidade relacionada com a categoria “Desporto”.

Assim, no `<a-mixin>` definiram-se algumas propriedades comuns a todas as categorias: a geometria do tipo *circle*, com raio de 1 metro; a formatação do texto; a escala; e o som de clique. Depois, na entidade de cada categoria, especificou-se: a posição, variável para cada entidade; a rotação, 10% no eixo do “x” e variável no eixo do “y” para cada entidade; e o texto a colocar. Atribuiu-se a componente *set-category*, referida em 5.5.6, o *mixin* respetivo e, ainda, dois eventos, para aumentar e repor a escala dependendo de o cursor estar ou não sobre a entidade. O resultado é visível na Figura 37.

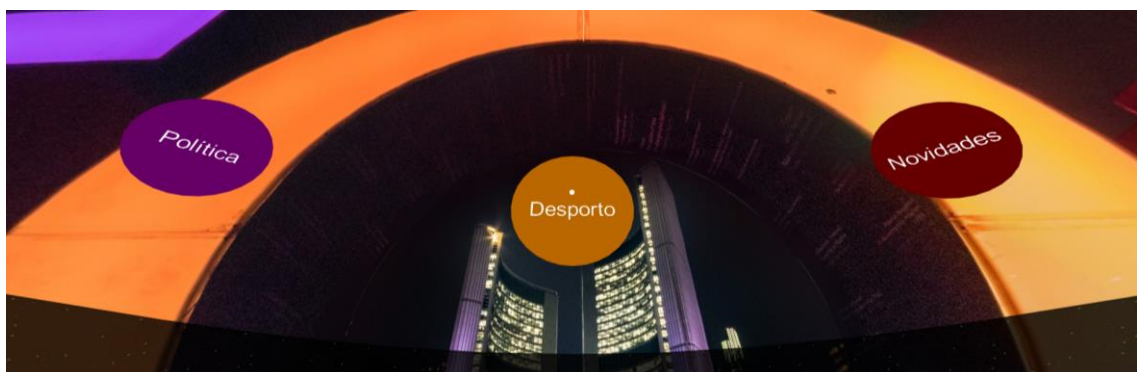


Figura 37 - Resultado visual das entidades relacionadas com as categorias.

5.6.7 Entidades relacionadas com a câmara e com o cursor

O componente da câmara define a perspetiva na qual o utilizador visualiza a cena. Esta está geralmente emparelhada com componentes de controlo, permitindo que *headsets* de realidade virtual ou telemóveis consigam mover e rodar a câmara. O componente do cursor fornece estados de deslocamento e de clique permitindo introduzir o elemento de interatividade.

Assim, para criar estes dois elementos na aplicação, desenvolveram-se três entidades. A primeira tem como função posicionar espacialmente a câmara; no caso definiu-se a posição (0,0,0). A segunda serve para adicionar a câmara e a componente *look-controls*, que permite rodar a câmara. A terceira é respetiva ao cursor e à sua configuração. Das propriedades do cursor, visíveis na Figura 38, destacam-se as seguintes:

- Clique após 3 segundos;
- Posicionamento a 3 metros de distância da câmara;
- Geometria do tipo *ring*, com raio interno de 0,0001 metros e raio externo de 0,02 metros;
- Material de cor branca;
- Animações para aumentar os raios interno e externo quando o cursor foca uma entidade clicável;
- Animações para repor os valores originais dos raios interno e externo quando o cursor deixa de focar uma entidade clicável;
- Animação para mudar a cor do cursor para vermelho quando este foca uma entidade clicável.

```
<a-entity position="0 0 0">
  <a-entity camera look-controls>
    <a-entity cursor="fuse: true; fuseTimeout: 3000"
      raycaster="objects: .clickable"
      position="0 0 -3"
      geometry="primitive: ring; radiusInner: 0.0001; radiusOuter: 0.02"
      material="color: white; shader: flat">
      <a-animation begin="fusing" attribute="geometry.radiusOuter"
        from="0.02" to="0.08" dur="10"></a-animation>
      <a-animation begin="fusing" end="mouseleave" delay="3000"
        attribute="geometry.radiusOuter" from="0.08" to="0.02" dur="10"></a-animation>
      <a-animation begin="mouseleave" attribute="geometry.radiusOuter"
        from="0.08" to="0.02" dur="10"></a-animation>
      <a-animation begin="fusing" attribute="geometry.radiusInner"
        from="0.0001" to="0.06" dur="10"></a-animation>
      <a-animation begin="fusing" end="mouseleave" delay="3000"
        attribute="geometry.radiusInner" from="0.06" to="0.0001" dur="10"></a-animation>
      <a-animation begin="mouseleave" attribute="geometry.radiusInner"
        from="0.06" to="0.0001" dur="10"></a-animation>
      <a-animation begin="fusing" end="mouseleave" easing="ease-in" attribute="material.color"
        fill="backwards" from="white" to="red" dur="3000"></a-animation>
    </a-entity>
  </a-entity>
</a-entity>
```

Figura 38 - Código das entidades relacionadas com a câmara e com o cursor.

5.7 Funcionamento

A aplicação possui dois modos de funcionamento: o modo *fullscreen*, para a utilização no computador, e o modo realidade virtual, para a utilização em conjunto com um *headset* próprio. Para ambos os casos, é necessário clicar no botão, indicado na Figura 39, para entrar nos respetivos modos de utilização.

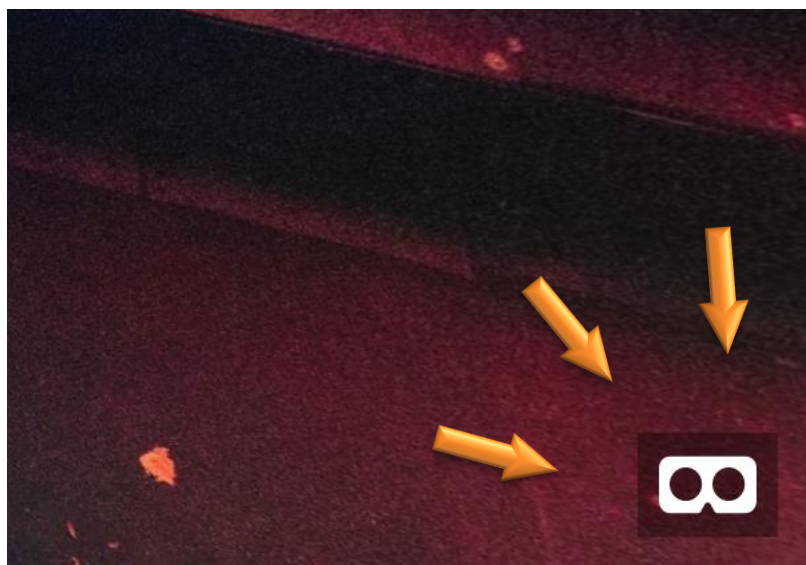


Figura 39 - Botão para entrar no modo *fullscreen* ou no modo realidade virtual com um *headset*.

De seguida, após ter entrado no modo de utilização pretendido, é possível escolher uma categoria, reproduzir e pausar um vídeo.

Relativamente ao sistema de reprodução/pausa dos vídeos, é explicado o seu funcionamento no gráfico da Figura 40.

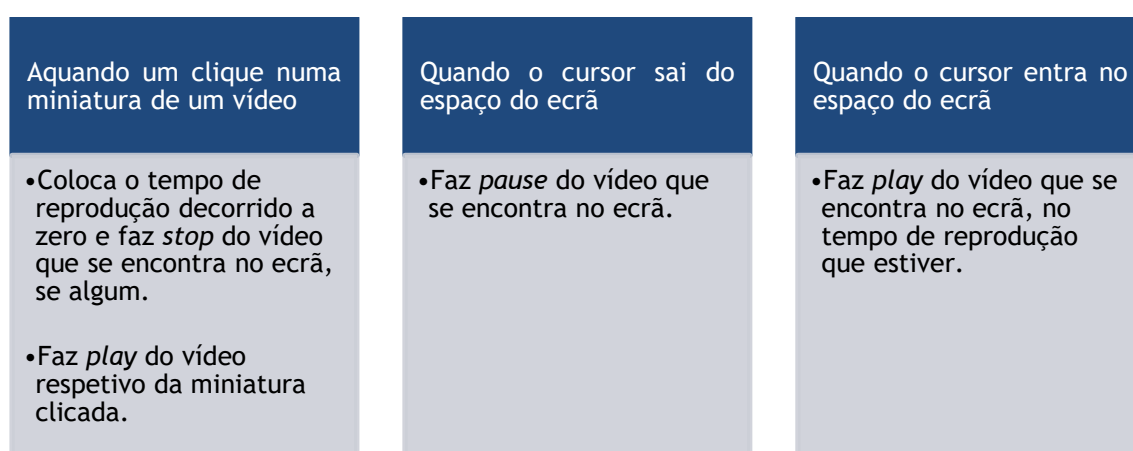


Figura 40 - Funcionamento do sistema de reprodução de vídeos.

5.7.1 Modo *fullscreen*

O modo *fullscreen* é obtido clicando no botão da Figura 39 para os utilizadores que pretendam abrir a aplicação no ecrã do computador. Este modo de utilização apenas mostra uma aplicação a 360 graus, não sendo o ideal para os utilizadores que queiram usufruir ao máximo da experiência que a aplicação tem para oferecer, uma vez que não possui a componente de imersão inerente à realidade virtual.

Desta forma, a interação do utilizador com a aplicação é feita através do clique e arrasto do rato.

5.7.2 Modo realidade virtual

O modo realidade virtual é o ideal para os utilizadores que queiram uma experiência imersiva. Este modo requer a utilização de um *headset* próprio. Para efeitos de testes, foi utilizado o *headset* representado na Figura 12, que corresponde a um *headset* da categoria *mobile*, ou seja, necessita da integração com um *smartphone*.

Assim, ao clicar no botão da Figura 39, deve-se obter um efeito parecido ao da Figura 41.

Relativamente à interação do utilizador com a aplicação, a navegação faz-se rodando o *headset* de realidade virtual enquanto que o clique é feito através de foco de 3 segundos num objeto clicável.

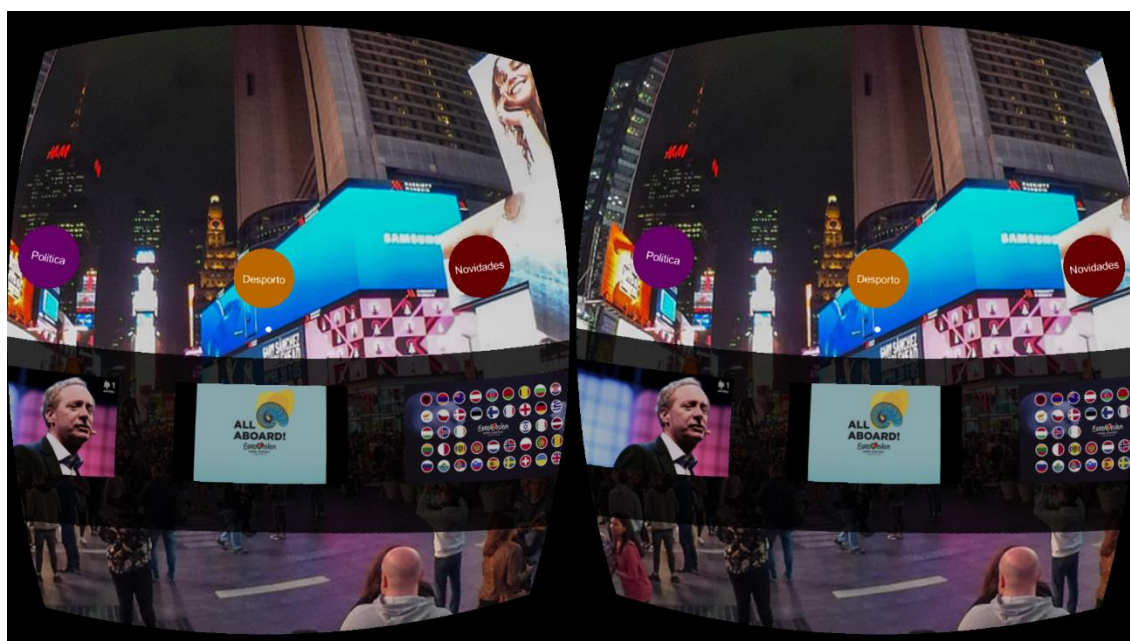


Figura 41 - Imagem estereoscópica criada no modo realidade virtual.

5.8 Problemas conhecidos

Durante o período de testes, detetaram-se alguns problemas na utilização da aplicação, que serão assinalados nas subsecções seguintes.

5.8.1 Problemas nos navegadores de *smartphone*

Os testes mais extensos foram efetuados no navegador Chrome, no entanto, também foram realizadas algumas experiências no navegador Firefox e no navegador nativo da Miui, o Mi Browser.

- Chrome

O Chrome não permite a reprodução automática de vídeos com som. É necessário chamar a função *play()* a um elemento HTML5 *audio*, através de um clique ou outra ação explícita do utilizador. Desta forma, a aplicação reproduz os vídeos, mas sem som.

- Firefox

No Firefox, não é possível reproduzir os vídeos nem os seus áudios.

- Mi Browser

No Mi Browser, à semelhança do Firefox, também não é possível reproduzir os vídeos nem os seus áudios.

5.8.2 Carregamento lento

Na primeira utilização da aplicação, o seu carregamento pode revelar-se um pouco lento, dependendo da ligação à internet que o utilizador esteja a usar. Isto deve-se ao grande número de ficheiros pesados presentes na aplicação e ao facto de ainda não existir no A-Frame um componente de pré-carregamento de recursos totalmente funcional.

5.8.3 Fundo preto

A cada 15 segundos a imagem de fundo é alterada. Por vezes, após algum tempo decorrido, o fundo fica preto, não ocorrendo mais trocas de imagens, sendo necessário atualizar a página para voltar ao funcionamento normal.

5.9 Discussão de resultados

Tendo sido desenvolvido um protótipo funcional, procedeu-se à realização de um questionário, possibilitando que um grupo de pessoas pudessem experimentar a aplicação com o objetivo de recolher os seus comentários e sugestões. Este método permitiu, ainda, reunir a opinião dos utilizadores acerca do futuro da realidade virtual e o seu impacto na *web*.

Deste modo, criou-se o seguinte questionário:

1 Idade

- <18
- 18-24
- 25-34
- 35-50
- >50

2 Está familiarizado(a) com sistemas de realidade virtual?

- Sim
- Não

3 Acredita que no futuro teremos uma *web* de realidade virtual interligada?

- Sim
- Talvez
- Não

4 Experimente agora a aplicação *web* por uns minutos. <http://vrmondoapp.ga>

De 0 a 5, classifique a sua experiência.

- 0
- 1
- 2
- 3
- 4
- 5

5 Comentários e sugestões (campo opcional)

6 Considera que uma plataforma deste género seria uma melhoria relativamente às plataformas já existentes para visualização de vídeos?

- Sim
- Talvez
- Não

Este questionário foi enviado para, potencialmente, 8059 estudantes da Faculdade de Engenharia da Universidade do Porto, dos quais foram obtidas 113 respostas, até à data da escrita deste documento.

Seguidamente, apresentam-se os gráficos que resumem as respostas recebidas.

Relativamente à pergunta 1, o resultado era o expectável, dado o público alvo a que se destinava o questionário. 85% dos inquiridos estão entre os 18 e os 24 anos (Figura 42).

Idade

113 respostas

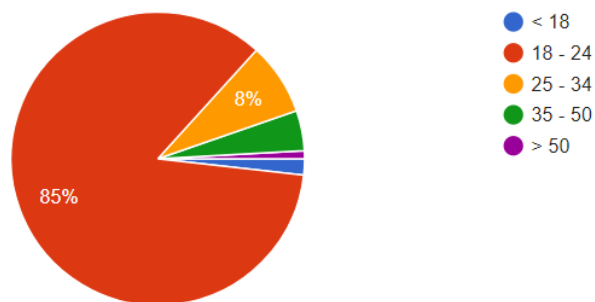


Figura 42 - Gráfico das respostas à pergunta 1.

Na pergunta 2, 75% dos inquiridos afirmaram já ter algum entrosamento com sistemas de realidade virtual (Figura 43). Este resultado é importante uma vez que estando a maioria dos inquiridos familiarizada com os sistemas de realidade virtual, possibilita resultados mais viáveis tendo em conta a aplicação apresentada.

Está familiarizado(a) com sistemas de realidade virtual?

113 respostas

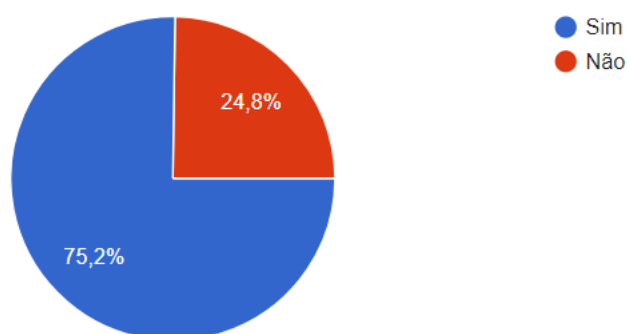


Figura 43 - Gráfico das respostas à pergunta 2.

Em relação à pergunta 3, 49,6% das pessoas acreditam que no futuro existirá uma *web* interligada em realidade virtual, ainda que 44,2% não tenham tanta certeza (Figura 44).

Acredita que no futuro teremos uma web de realidade virtual interligada?

113 respostas

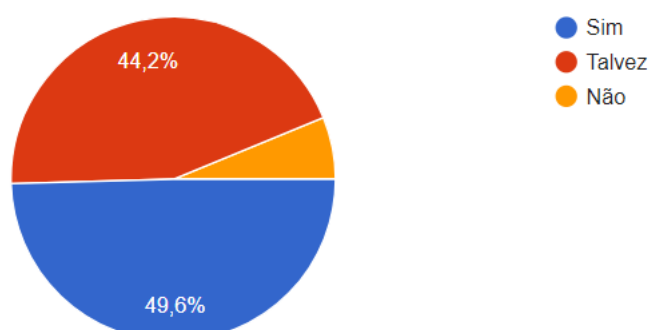


Figura 44 - Gráfico das respostas à pergunta 3.

Quanto à classificação da sua experiência após utilizarem a aplicação, a maioria dos inquiridos dividiu-se entre o valor três e quatro (Figura 45). A classificação média, arredondada às décimas, situa-se nos 3,3.

De 0 a 5, classifique a sua experiência.

113 respostas

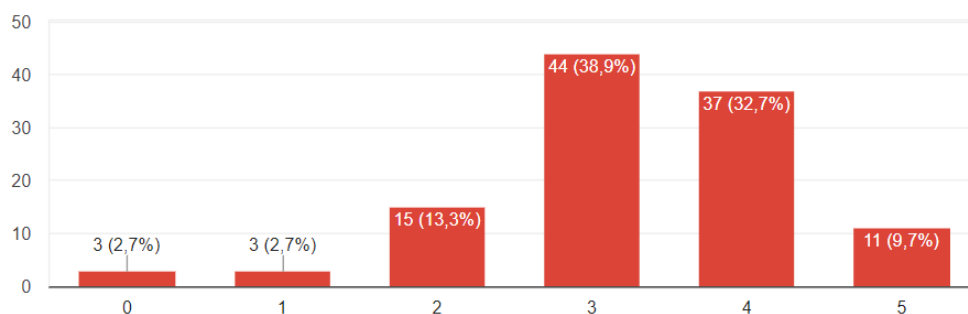


Figura 45 - Gráfico das respostas à pergunta 4.

Relativamente à pergunta 5, foi solicitado aos inquiridos que deixassem um comentário ou sugestão da sua experiência. Foram recebidas 23 respostas. Seguidamente apresentam-se os tópicos dos temas mais referenciados:

- Cinco pessoas referiram algum desagrado com o modo de navegação da aplicação no computador: ser necessário clicar para arrastar a imagem, não permitir mover a câmara mais rapidamente e mudar direção de arrasto foram as mais mencionadas.
- Duas pessoas indicaram a ausência de instruções de como utilizar a aplicação.
- Duas pessoas referiram que o ecrã onde se reproduzem os vídeos deveria ser maior (modo *fullscreen* no computador).
- Duas pessoas mencionaram o tempo de espera na abertura da aplicação.

Também foram obtidos alguns comentários de agrado:

- “Como primeira experiência, gostei. Utilizei o Ms Edge, com ligação em fibra até ao router e rede gigabyte até ao computador. Experiência fluída, de tal forma que inicialmente existe alguma desorientação. Mas acredito num desenvolvimento deste tipo de projetos.”
- “Ideia interessante.”
- “Bom trabalho, camarada!”

Estas sugestões e comentários são muito importantes para refletir acerca do trabalho desenvolvido com vista a poder melhorar a aplicação e progredir, tanto neste trabalho como futuramente, enquanto profissional.

Por último, na pergunta 6, 46% dos inquiridos não têm uma opinião marcada sobre se uma aplicação deste género representaria uma melhoria em relação às existentes, enquanto que 40,7% afirmaram que sim (Figura 46). É de ressaltar que apenas 13,3% responderam negativamente. Este é um resultado muito significativo que permite afirmar que esta é uma aplicação de qualidade e que uma aplicação idêntica teria utilidade.

Considera que uma plataforma deste género seria uma melhoria às plataformas já existentes para visualização de vídeos?

113 respostas

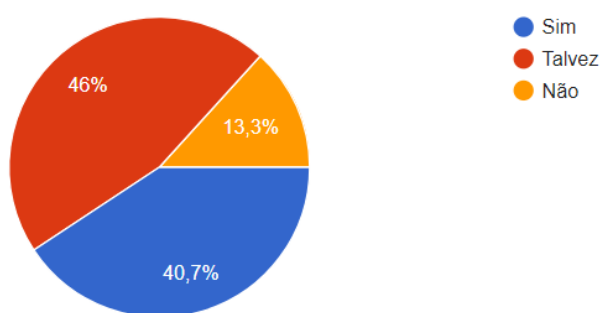


Figura 46 - Gráfico das respostas à pergunta 6.

Capítulo 6

Conclusões e trabalho futuro

Neste capítulo são apresentadas as principais conclusões obtidas na realização desta dissertação e alguns possíveis desenvolvimentos futuros.

6.1 Conclusões

Esta dissertação abordou a temática da visualização de conteúdos multimédia e explorou o potencial que a realidade virtual tem para oferecer, particularmente a componente imersiva. Assim, o trabalho realizado neste projeto consistiu no desenvolvimento de uma solução que inovasse a forma como conteúdos são consumidos pelo utilizador.

Foi possível concluir que o A-Frame é uma excelente ferramenta para a criação de aplicações em WebVR, e que, apesar de fácil de utilizar, é uma *framework* poderosa com uma das maiores comunidades de realidade virtual.

Em relação à aplicação concebida, foi desenvolvido um protótipo funcional, interativo e inovador, os três requisitos identificados na apresentação da solução proposta. Foram também efetuados vários testes, tendo-se encontrado alguns problemas, nomeadamente na utilização com os navegadores *mobile*.

Por fim, procedeu-se à realização de um questionário que permitiu que um grupo de pessoas experimentassem e dessem a sua opinião quanto à aplicação desenvolvida, tendo-se obtido uma boa classificação da sua experiência.

6.2 Trabalho futuro

Tendo como objetivo o melhoramento da aplicação sugerem-se vários tópicos para investigação e desenvolvimento futuros. Para além da resolução dos problemas conhecidos referidos na secção 5.8, são exibidas na Figura 47 algumas propostas de como a aplicação poderia evoluir. Neste trabalho foi abordada a Fase 1. Poder-se-iam seguir as Fase 2 e 3 que permitiriam aumentar os recursos disponíveis para utilização e adicionar a inteligência relacionada com a personalização.

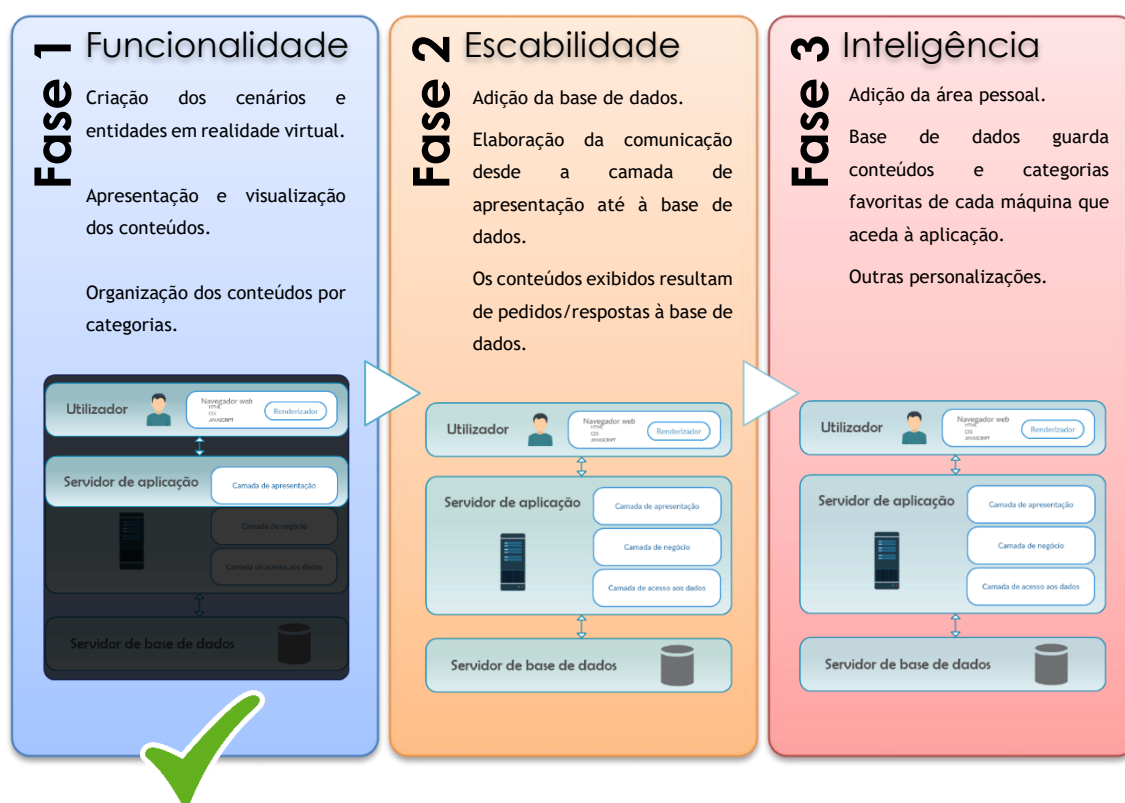


Figura 47 - As 3 fases de trabalho da aplicação idealizada.

Referências

- [1] Tec4Growth - INESCITEC, “FourEyes Technical Annex”.
- [2] CogniFit, “Percepção,” [Online]. Available: <https://www.cognifit.com/pt/percecao>. [Acedido em 19 abril 2017].
- [3] C. Ware, Information Visualization - Perception for Design, 3ª ed., Morgan Kaufmann, 2012.
- [4] M. Lima, “A visual history of human knowledge,” março 2015. [Online]. Available: https://www.ted.com/talks/manuel_lima_a_visual_history_of_human_knowledge?utm_source=tedcomshare&utm_medium=referral&utm_campaign=tedspread. [Acedido em 2 maio 2017].
- [5] D. McCandless, “A beleza da visualização de dados,” julho 2010. [Online]. Available: https://www.ted.com/talks/david_mccandless_the_beauty_of_data_visualization?language=pt. [Acedido em 19 abril 2017].
- [6] V. Mahinder, “2D vs 3D Animation—Pros and Cons,” [Online]. Available: <https://medium.com/@vandhana.appiness/2d-vs-3d-animation-pros-and-cons-51daba16a4ec>. [Acedido em 19 maio 2017].
- [7] Wikipedia, “WebVR,” [Online]. Available: <https://en.wikipedia.org/wiki/WebVR>. [Acedido em 13 outubro 2017].
- [8] M. Wilson, “Why A Virtual Reality Web May Never Happen,” 4 maio 2016. [Online]. Available: <https://www.fastcodesign.com/3058591/why-a-virtual-reality-web-may-never-happen>. [Acedido em 13 outubro 2017].
- [9] A. Paracuellos, “Do Websites Dream of Virtual Reality?,” 10 dezembro 2016. [Online]. Available: <https://www.awwwards.com/do-websites-dream-of-virtual-reality.html>. [Acedido em 13 outubro 2017].
- [10] E. Olshannikova, A. Ometov, Y. Koucheryavy e T. Olsson, “Visualizing Big Data with augmented and virtual reality: challenges and research agenda,” *Journal of Big Data*, p. 10, 2015.

- [11] Wikipedia, “Chart,” [Online]. Available: <https://en.wikipedia.org/wiki/Chart>. [Acedido em 19 maio 2017].
- [12] Wikipedia, “Information visualization,” [Online]. Available: https://en.wikipedia.org/wiki/Information_visualization. [Acedido em 19 maio 2017].
- [13] Wikipedia, “Graph drawing,” [Online]. Available: https://en.wikipedia.org/wiki/Graph_drawing. [Acedido em 19 maio 2017].
- [14] Interaction Design Foundation, “How to Design an Information Visualization,” [Online]. Available: <https://www.interaction-design.org/literature/article/how-to-design-an-information-visualization>. [Acedido em 13 outubro 2017].
- [15] R. Mazza, Introduction to Information Visualization, Springer, 2009.
- [16] Khronos Group, “WebGL - OpenGL ES for the Web,” [Online]. Available: <https://www.khronos.org/webgl/>. [Acedido em 10 maio 2017].
- [17] Wikipedia, “WebGL,” [Online]. Available: <https://en.wikipedia.org/wiki/WebGL>. [Acedido em 10 maio 2017].
- [18] F. Boesch, “Why you should use WebGL,” fevereiro 2013. [Online]. Available: <http://codeflow.org/entries/2013/feb/02/why-you-should-use-webgl/>. [Acedido em 10 maio 2017].
- [19] Wikipedia, “Unity,” [Online]. Available: <https://pt.wikipedia.org/wiki/Unity>. [Acedido em 10 maio 2017].
- [20] J. Gomez, “Unity 5 WebGL vs Web Player. 12 Pros and Cons,” 28 agosto 2015. [Online]. Available: <http://unitydojo.blogspot.pt/2015/08/unity-5-webgl-vs-web-player-12-pros-and.html>. [Acedido em 10 maio 2017].
- [21] Wikipedia, “Stage3D,” [Online]. Available: <https://en.wikipedia.org/wiki/Stage3D>. [Acedido em 10 maio 2017].
- [22] D. Jackson, “Next-generation 3D Graphics on the Web,” 7 fevereiro 2017. [Online]. Available: <https://webkit.org/blog/7380/next-generation-3d-graphics-on-the-web/>. [Acedido em 10 maio 2017].
- [23] Wikipedia, “Canvas element,” [Online]. Available: https://en.wikipedia.org/wiki/Canvas_element. [Acedido em 30 maio 2017].
- [24] Wikipedia, “Three.js,” [Online]. Available: <https://en.wikipedia.org/wiki/Three.js>. [Acedido em 10 maio 2017].
- [25] G. Stier, “WebGL and A-Frame,” 7 junho 2016. [Online]. Available: <https://solutiondesign.com/blog/-/blogs/63714>. [Acedido em 10 maio 2017].
- [26] T. Authors, “three.js - Javascript 3D library,” [Online]. Available: <https://threejs.org/>. [Acedido em 10 maio 2017].

- [27] R. Upadhyaya, “3D GRAPHICS WITH WEBGL, THREE.JS & BABYLON.JS,” 3Pillar Global, [Online]. Available: <https://www.3pillarglobal.com/insights/3d-graphics-with-webgl-three-js-babylon-js>. [Acedido em 10 maio 2017].
- [28] B. Authors, “BabylonJS - 3D engine based on WebGL/Web Audio and JavaScript,” [Online]. Available: <https://www.babylonjs.com/>. [Acedido em 10 maio 2017].
- [29] Wikipedia, “A-Frame (virtual reality framework),” [Online]. Available: [https://en.wikipedia.org/wiki/A-Frame_\(virtual_reality_framework\)](https://en.wikipedia.org/wiki/A-Frame_(virtual_reality_framework)). [Acedido em 10 maio 2017].
- [30] Mozilla, “A-Frame - Make WebVR,” [Online]. Available: <https://aframe.io/>. [Acedido em 10 maio 2017].
- [31] Wikipedia, “PlayCanvas,” [Online]. Available: <https://en.wikipedia.org/wiki/PlayCanvas>. [Acedido em 15 maio 2017].
- [32] bnjm, “WEBGL FRAMEWORK COMPARISON,” [Online]. Available: <http://bnjm.github.io/WebGL-framework-comparison/>. [Acedido em 15 maio 2017].
- [33] PLAYCANVAS, “PlayCanvas - 3D HTML5 & WebGL Game Engine,” [Online]. Available: <https://playcanvas.com/>. [Acedido em 15 maio 2017].
- [34] Wikipedia, “List of WebGL frameworks,” [Online]. Available: https://en.wikipedia.org/wiki/List_of WebGL_frameworks. [Acedido em 15 maio 2017].
- [35] M. Bostock, “D3.js - Data-Driven Documents,” [Online]. Available: <https://d3js.org/>. [Acedido em 17 maio 2017].
- [36] Wikipedia, “D3.js,” [Online]. Available: <https://en.wikipedia.org/wiki/D3.js>. [Acedido em 17 maio 2017].
- [37] Wikipedia, “Comparison of JavaScript charting frameworks,” [Online]. Available: https://en.wikipedia.org/wiki/Comparison_of_JavaScript_charting_frameworks. [Acedido em 17 maio 2017].
- [38] S. Authors, “Sigma js,” [Online]. Available: <http://sigmajs.org/>. [Acedido em 17 maio 2017].
- [39] A. B.V., “vis.js - A dynamic, browser based visualization library.,” [Online]. Available: <http://visjs.org>. [Acedido em 17 maio 2017].
- [40] graphAlchemist, “Alchemy.js,” [Online]. Available: <http://graphalchemist.github.io/Alchemy/#/>. [Acedido em 17 maio 2017].
- [41] L. Sarmento, F. Gouyon, B. G. Costa e E. Oliveira, “Visualizing networks of music artists with RAMA,” *Artigo em Livro de Atas de Conferência Internacional*, 2009.
- [42] A. S. Rose e P. W. Hildebrand, “NGL Viewer: a web application for molecular visualization,” *Nucleic Acids Research*, vol. 43, p. W576-W579, 2015.

- [43] A. S. Rose e P. W. Hildebrand, “NGL,” [Online]. Available: <http://proteininformatics.charite.de/ngl-tools/ngl/html/ngl.html>. [Acedido em 13 outubro 2017].
- [44] Kinetika , “Silk River,” [Online]. Available: <http://www.silkriver.co.uk/>. [Acedido em 13 outubro 2017].
- [45] L. Scott, “Silk River (web app),” *Bath School of Art and Design Library*, 2017.
- [46] Kinetika, “Silk River (web app),” [Online]. Available: <http://www.silkriver.co.uk/app/>. [Acedido em 13 outubro 2017].
- [47] J. Strietelmeier, “VR Shinecon Virtual Reality Glasses review,” 11 janeiro 2016. [Online]. Available: <https://the-gadgeteer.com/2016/01/11/vr-shinecon-virtual-reality-glasses-review/>. [Acedido em 9 janeiro 2018].
- [48] Mozilla VR, “A-Painter: Paint in VR in Your Browser,” [Online]. Available: <https://blog.mozvr.com/a-painter/>. [Acedido em 12 janeiro 2018].
- [49] A-Frame, “<a-sky>,” [Online]. Available: <https://aframe.io/docs/0.7.0/primitives/a-sky.html>. [Acedido em 20 janeiro 2018].
- [50] A-Frame, “<a-curvedimage>,” [Online]. Available: <https://aframe.io/docs/0.7.0/primitives/a-curvedimage.html>. [Acedido em 20 janeiro 2018].
- [51] Awwwards Magazine, “Do Websites Dream of Virtual Reality?,” [Online]. Available: <https://www.awwwards.com/awards/gallery/2015/12/vr-interfaces-awwwards-unboring-2.png>. [Acedido em 9 janeiro 2018].
- [52] A-Frame, “A-Frame Registry,” [Online]. Available: <https://aframe.io/aframe-registry/>. [Acedido em 21 janeiro 2018].